

THESIS

COMPUTERIZED SPACE FLIGHT PLANNING

Submitted by
Craig Matthew Brown

In partial fulfillment of the requirements
for the Degree of Master of Science
Colorado State University
Fort Collins, Colorado

June, 1973

COLORADO STATE UNIVERSITY LIBRARIES

TL790
B76

end

ENTITLED Computerized Space Flight Planning

Committee on Graduate Work

William C Boes

Sanford B. Thayer
Adviser

Fred W. Smith
Head of Department

ABSTRACT

COMPUTERIZED SPACE FLIGHT PLANNING

This thesis analyses manual and computerized techniques of preparing timelines of crew activities and orbital parameters in support of the NASA Skylab program. A brief discussion of timelines and their use in mission analysis is provided. Procedures used in both the manual and computerized generation of mission timelines are discussed and computer programs provided. Early program evaluation and current hardware requirements are discussed as they relate to the timeline generation task.

An indepth cost-effectiveness study of flight planning alternatives is conducted to establish the superiority of one of the alternatives in meeting cost-effectiveness goals. Cost and effectiveness parameters are created and used in the analysis as a means of quantitatively measuring each alternatives cost-effectiveness. Engineering economy procedures for calculating a rate of return are applied to the computerized alternative and the results analyzed.

Finally, an overall systems cost-effectiveness relative value of merit is established for each alternative using a ratio cost-effectiveness model. This value of merit is then used to establish the cost-effectiveness superiority of one of the alternatives. Conclusions about the alternatives are

then drawn from the analyses and recommendations made as to their implementation for Skylab mission analysis.

Craig Matthew Brown
Department of Mechanical Engineering
Colorado State University
Fort Collins, Colorado 80521
June, 1973

TABLE OF CONTENTS

<u>Chapter</u>		<u>Page</u>
1	INTRODUCTION AND OBJECTIVES.	1
	A. Introduction	1
	B. Objective.	1
	C. Brief Description of Skylab Program. .	2
2	COMPUTERIZED TIMELINE SUPPORT.	6
	A. Description of Mission Timeline and Their Use.	6
	B. Old Manual Method Verses Computerized Construction of Timelines.	7
	C. Description of Flight Planning Computer Programs and What They Accomplish. .	8
	D. Program Evolution and Computer Systems Used in Running Programs	15
3	COST-EFFECTIVENESS ANALYSIS.	19
	A. Approach to Cost-Effectiveness Analysis	19
	B. Cost-Effectiveness Analysis.	22
	C. Engineering Economy Analysis	42
	D. Systems Expected Value	50
4	CONCLUSIONS AND RECOMMENDATIONS.	52
	Bibliography	54
	Appendix A - Detailed Discussion of Plotting Programs.	56
	Appendix B - Training Manual	73
	Appendix C - Cost-Effectiveness Support Calculations	89
	Appendix D - NEWTIME Program	91
	Appendix E - NEWCREW Program	106

LIST OF TABLES

	<u>Page</u>
Table 1 - Developmental and Operational Costs Associated with Computerized Generation.	41
Table 2 - Cash Flow Before Taxes.	43
Table 3 - Cash Flow Before Taxes - 10% Computer Overrun.	45
Table 1B - Appendix B Table of Days Between Two Dates	79
Table 2B - Appendix B Table of New Moons . . .	81

LIST OF FIGURES

	<u>Page</u>
Figure 1 - Manually Constructed Timeline Frame. .	10
Figure 2 - Final Result of NEWTIME Plotting . . .	12
Figure 3 - Final Result of NEWTIME and NEWCREW Plotting	14
Figure 4 - Domain of Feasibility Study.	37

CHAPTER 1

INTRODUCTION AND OBJECTIVES

Introduction

This thesis analyses competitive methods of generating mission timelines in support of the NASA Skylab program. Since Skylab is a mission designed to gather as much information as possible in a relatively short period of time about man's performance in space, the emphasis has been placed on efficiently planning the crew's time for the manned portions of the mission. A planning aid, known as a mission timeline, was developed to provide the crew with an easy way to see which activities must be performed at various stages of the mission. In addition to aiding the crew, it is also designed to give the flight planners a means of effectively planning the tasks that must be completed by the crew.

A technical discussion of a manual method and a computerized method of timeline generation will be conducted. Procedures used in implementing each alternative will also be discussed in detail. A cost-effectiveness study will be used to compare the alternatives and cost and effectiveness parameters will be established to support this study. The methodology used to establish the necessary cost and effectiveness parameters will be developed and a logical flow of analysis will be used to draw conclusions about the systems.

Objective

It is the objective of this thesis to develop and apply cost-effectiveness criteria applicable to a decision to

computerize the construction of timelines. The procedures and techniques used in this thesis will be flexible enough to then be employed to analyze any competitive alternatives on a cost-effectiveness basis.

Brief Description of the Skylab Program

Most Americans and many people throughout the world are familiar with the United States manned accomplishments in space achieved through projects Mercury, Gemini and Apollo, but few know much about the Skylab Program that is to be the successor to the Apollo effort. In the near future the emphasis in space exploration will shift from lunar exploration to near earth orbit study missions. To reach this goal the National Aeronautics and Space Administration (NASA) has established the Skylab Program for the purpose of satisfying the requirements of near earth study missions. Skylab, as it is presently defined, is actually four separate missions - Skylab I through Skylab IV.

Skylab I - The only unmanned segment of the program involves launching an orbital workshop (OWS) into an earth orbit. This orbital workshop is actually a Saturn IV-B that has been modified and converted into an orbital space laboratory. The workshop is completely equipped with various hardware packages that will form the nucleus of the experiments that are to be conducted on Skylabs II, III, and IV. The experiments planned for Skylab are designed to study three main categories of interest in near earth orbits.

The first category is medical experiments that are designed to test man's physiological adaptation to working under zero gravity conditions. It is hoped that the results of this category of experiments will produce much useful baseline data that can be used in planning future space programs such as manned planetary exploration and space stations.

Category II consists of scientific experiments designed to provide researchers with useful information concerning the earth and the universe in which we live. Experiments, such as the Earth Resources Experiment Package (EREP) and celestial viewing experiments, are designed to study the earth and our universe in a way that it never before has been studied, free from the constraints such studies are subject to on earth.

The third category of experiments are engineering experiments that are designed to establish what man can and cannot do while working in zero gravity. The aim of these experiments is to establish limits to man's working capabilities while under zero gravity constraints. It is hoped that these experiments will provide designers of future missions with such information as the type of maintenance tasks that can be performed by man in zero gravity.

Skylab II - Once the OWS is in orbit the first manned mission, Skylab II, will be launched with three crewmen aboard to dock with the workshop. The current plans call for

the three crewmen to spend 28 days in orbit. The early phases of the mission require the crew to enter the OWS, check to insure all systems are operable and prepare the workshop for habitability by the crew. Once the internal environmental control systems have been activated it will be possible for the crew to work within the lab in a shirtsleeved environment without the necessity for pressurized space suits. When the initial activation of the lab is complete, the crew will be called upon to conduct some of the experiments aboard before returning to earth. When the crew of Skylab II returns to earth they will leave all of the environmental control systems aboard the workshop partially activated in preparation for the arrival of the three man crew of Skylab III.

Skylabs III & IV - Skylab III is a 56 day mission whose objective is to conduct the medical, scientific and engineering experiments aboard the OWS and to assess the capabilities of man to perform after being exposed to a 56 day period of zero gravity. At the conclusion of Skylab III a similiar 56 day Skylab mission, Skylab IV, will be launched to conduct further studies in the OWS and at its conclusion the OWS will be deactivated, the crew will return to earth and the Skylab, as it is presently defined, will terminate.

At the conclusion of Skylab IV the United States will have accumulated data about man's capability to perform in space. Such data about long duration mission has never

before been available. A total of 140 days in space will be accumulated and a solid baseline for future missions will be established.

CHAPTER 2

COMPUTERIZED TIMELINE SUPPORT

Description of Mission Timelines and Their Use

Since Skylab is a mission designed to gather as much information as possible in a relatively short period of time about man's performance in space, a large emphasis has been placed on efficiently planning the crews time for manned portions of the mission. In order to accomplish this a planning aid, known as a timeline, was developed to give the planners a means of effectively planning the tasks which must be completed by the crew. It is designed to give the crew and ground personnel an easy way to see which activities must be performed at various stages of the mission.

In planning the activities for any given period of time, the flight planner must schedule certain activities to meet various constraints that restrict the performance of such tasks. For example, it would not be desirable to schedule an experiment that is designed to photograph the earth while the spacecraft is orbiting in the earth's shadow. Therefore the timeline was created to give the flight planner a chronological display of when certain orbit dependent occurrences took place. These orbital dependent parameters were such occurrences as correlation between different time scales used to time the mission Ground Elapsed Time (GET) verses Greenwich Mean Time (GMT). Other orbit related parameters were then related to these time scales as they occurred ie, times of orbital light and darkness, times over a radiation belt known as the South Atlantic Anomaly (SAA),

times over various tracking stations that comprise the Spaceflight Tracking and Data Network, times over various continents and sun angles at the beginning of each time period. This information could then be used by the flight planner to schedule activities within the limits of various orbital constraints.

The current operating procedure allows the flight planner to have the activities he has scheduled entered onto the timeline in the proper periods that they are to occur. Any scheduling errors become apparent in the form of overlapping activities or gaps between activities. An easy correlation can be made between crew activities and orbit dependent parameters ie, a crewman knowing his present time in the mission will be able to use the flight plan to determine when future events will occur such as when he will be over a certain tracking station or when he will enter the darkness portion of the orbit.

Old Manual Method Verses Computerized Construction of Timelines

In the early days of the Skylab Program the timeline portion of the flight plan was constructed by hand, using draftsmen to draw the basic form and having engineers hand fit the orbit dependent parameters onto the frame in the proper time reference locations. As long as the requirements for the number of orbit parameters remained small this technique produced satisfactory results.

As more and more parameters were required by the flight planner and the crew, the manual construction of the timeline frames and the hand fitting of data became impracticable. It required a large number of man hours for draftsmen to hand draw enough frames for the flight planner to use and the accuracy of the orbital parameters that were entered left much to be desired. It was becoming more difficult for the flight planners to meet contractual delivery dates and therefore a decision was made to study alternative ways of completing the timelines.

A study was performed to assess the feasibility of using graphical hardware tools available in conjunction with computer programs to construct the frames, enter the orbital dependent information on the frames and even enter the planners activities onto the frames once the mission is planned. A recommendation was made to pursue the course of computerization and it is the purpose of this thesis to examine whether this decision was a cost-effective one or not.

Description of Flight Planning Computer Programs and What They Accomplish

Prior to discussing the computerized techniques employed to generate the timeline, it may be helpful to briefly discuss the manual procedures that were employed prior to the introduction of computerization.

In the early days of the Skylab Program a need was recognized for graphically displaying data concerning orbital parameters and crew activities on a time scale so that various occurrences during the mission could be studied and planned chronologically. The means used to accomplish this task was the mission timeline. In its early form it was no more than a 24 hour time scale with entries made chronologically to depict crew activities and a few basic orbit dependent parameters such as day-night cycles and tracking station coverage. Computer programs, executed at the L. B. Johnson Space Center, Houston, Texas, generated information in printout form that stated times these various parameters occurred. Draftsmen, using standard drafting room techniques, would hand construct a basic form that contained the 24 hour time scale with provisions for entering crew activities and orbit dependent parameters. These frames were then given to engineers who, using the NASA printouts of orbital parameters, hand fit the data onto the frame in proper chronological sequencing. This hand fitting of data was a very tedious task and in many cases was a source of inaccurate entries. Once this form construction and data fitting was complete, the timelines were given to the flight planners who then planned the crew activities conforming to any constraints that might be imposed due to orbital conditions. Many times, due to the inherent simplicity of the manually constructed frame, the flight planner was forced to consult additional sources to check orbital constraints

Day of Mission		Day 1 of 29 Day Mission			
Day / Night		Day	Day	Day	Day
Ground Station Coverage		<u>BDA</u>		<u>CRO</u>	<u>GWM</u>
		<u>ACN</u>	<u>HAW</u>		<u>HAW</u>
		<u>TAN</u>		<u>TAN</u>	
		<u>ANG</u>	<u>SWE</u>		<u>HWI</u>
Revolution		1 ↑	2 ↑		3 ↑
GMT(M/D/H) 10/02/		22 ↑	23 ↑	00 ↑	01 ↑
Crew Activities	1	Statkp	Dock	526 ML	Syshk
	2	Statkp	Dock	526 ML	Activn
	3	Statkp	Dock	526 ML	Activn
Exper. Sys. Req. Vehicle Orient & Event Sequence		OA NXP-Z		OA NX-POP	
		Sta. Keep	CSM Dock	AM-MDA Activ	
D/H After L/O (GET)00/04		00/05	00/06	00/07	00/08

Figure 1 - Manually Constructed Timeline Frame

that were not depicted on the timeline frame. Once the flight planner had completed his task, his planning information along with the frames were returned to the draftsmen who entered the activities according to the flight planners instructions. This human intervention, as in the case of entering orbital parameters, was a source of many inaccuracies. Errors occurred due to the tediousness of entering large amounts of data by hand. There were times when blocks of orbital parameters were shifted in time or entered entirely on the wrong day. Figure 1 illustrates a sample of an earlier version of a manually constructed timeline frame.

As the requirements for the amount of data displayed increased and accuracy became more critical it was decided that possibly the computerization of the flight planning task would be beneficial. As it exists today the task, except for the continued need for the flight planners intervention to schedule crew activities, has become completely computerized. The critical data that was previously supplied from NASA in the form of printouts is now also supplied in the form of computer magnetic tapes. The data on these tapes is much broader in scope and contains many more parameters than the earlier information supplied in the form of printouts for manual incorporation.

Two graphical support programs were developed to accept the NASA generated magnetic tapes as input and generate plotted output. A brief discussion of these two computer

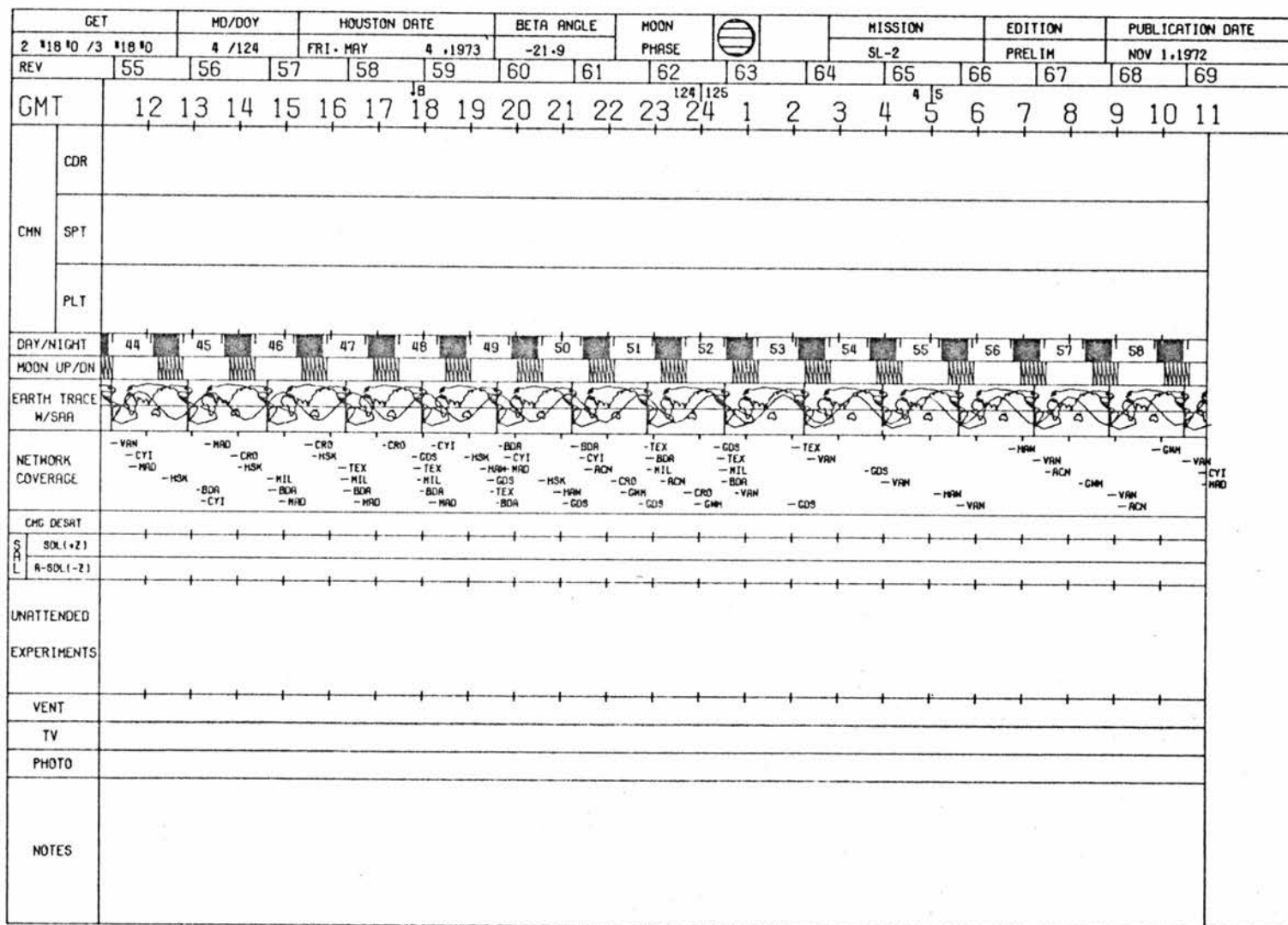


Figure 2 - Final Result of NEWTIME Plotting

programs - NEWTIME and NEWCREW - will follow with a more detailed discussion available in Appendix A.

The support program responsible for the construction of the basic timeline frame and the incorporation of trajectory dependents variables is NEWTIME. Trajectory changes, dictated by changes in the mission, require regeneration of these basic input tapes. For every trajectory change, a new set of tapes is required as input to the NEWTIME program. In order to extract the pertinent data from each of these tapes, reduction programs exist to pull the necessary data from the tapes and reformat it so that it is in the proper form to be used as input to the NEWTIME program.

A sample timeline plot of the type generated in the NEWTIME program is shown in Figure 2. At this point a correlation may be established for all of the various parameters that are occurring at any chosen time. This feature is what the flight planner depends on to be able to effectively plan the activities for the mission.

At this point an entire mission of timeline frames are plotted, with a 28 day Skylab II mission requiring about 29 or 30 frames including launch and splashdown frames. A 56 day Skylab III or IV mission requires approximately 57 to 59 frames.

These complete frames are then turned over to the flight planners who are responsible for planning the activities of the crew and the experiments to be performed within various



PH-PERSONAL HYGIENE
PLN-PLANNING
PLT-PILOT
PT-EXERCISE
REV-REVOLUTION
R&R-REST AND RECREATION
SAA-SOUTH ATLANTIC ANOMALY
SAL-SCIENTIFIC AIRLOCK
S/HK-SYSTEMS HOUSEKEEPING
SOL-SOLAR
SPT-SCIENTIST PILOT
ST-STOW
TEX-TEXAS
VAN-VANGUARD

Figure 3 - Final Result of NEWTIME and NEWCREW Plotting

constraints. The activities are penciled onto the frames in the proper times that they are to occur. Any adjustments that are necessary to accomodate various experiments are made and when the final acceptable version is reached the second and final program that overlays the crew activities along with momentum dump information, scientific airlock information and unattended experiment information is ready to prepare to run. This program - NEWCREW - allows the flight planner to use a cathode ray remote terminal to create data files that describe the activities he wishes to plot in the correct time period he wishes to plot them.

A sample of a final plot that is the result of plotting done with the NEWTIME and NEWCREW program is shown in Figure 3. This figure also gives a brief description of the contents of the timeline frame. Appendix A maybe used for more detailed information on frame content.

Program Evolution and Computer Systems Used in Running Programs

When development of the programs began, the only computer system available with the CalComp software package which is necessary for plotting was an IBM 360/30. In the early phases of the program not all of the data requested by the flight planners was available on the NASA tape that was received from Houston. Therefore it was necessary to execute a series of preliminary trajectory programs that calculated many orbital parameters necessary as input to the plotting routines. These preliminary programs were executed on a Control Data Corporation 6400-6500 series computer system.

They required a considerable amount of time preparing the programs to run. Trajectory data matching was required and thus the results of the preliminary programs often were not as accurate as the real trajectory data. Once the programs had been prepared and executed, it was necessary to use the output, which was in the form of punched cards and magnetic tapes, as input to the plotting programs on the IBM 360/30. A hardware limitation of three available tape drives and only one card reader prevented any expansion of data that could have been presented on the frames in the future.

Eventually the necessary CalComp software package became available on the Control Data Corporation 6400-6500 series computers and at the same time NASA increased the amount and quality of information supplied in their trajectory tapes. The compatibility of the faster CDC equipment with plotting programs and the fact that the cumbersome preliminary trajectory programs no longer were needed, improved the quality and speed of turn around time for the output.

In addition to the added efficiency of the CDC 6400-6500 computers, a remote terminal operating system known as the File Alteration Storage Terminal (FAST) system was available for the conversion and expansion necessary when the programs were removed from the IBM 360/30 and placed on the CDC 6400-6500. The FAST system allows extremely rapid formation of control, source and data files. It is also possible to make changes or alterations to a program much more efficiently than if computer punched cards had to be created on a

keypunch. This system also allows a user to execute a program from the remote terminal, follow the execution of the program throughout the various stages of execution and examine the output immediately upon its availability in the output queue. At this point any problems or errors may be corrected from the terminal and the program re-executed. Any changes necessitated by user requests can also be more rapidly incorporated through the use of FAST than if punched cards were used.

The plotting hardware used throughout the developmental stages and currently in the operational stage, is the CalComp model 763 drum plotter used in conjunction with the CalComp model 770 tape drive.

Due to the nature of the usage of the output from these programs there is a requirement that, as the Skylab program approaches an operational status, the programs developed and operational at Martin Marietta's Denver Division be operationally compatible at the L. B. Johnson Space Center, Houston, Texas. Various trips have been made to the Space Center to assess the compatibility of the programs and it has been determined that, with slight modifications, the programs could be made operational at either sight. The CalComp software package exists on a CDC 6400 operating system at the Space Center. This system is currently used in conjunction with two command module simulators used to train astronauts for the Skylab program. The plotting programs have been run on the NASA computers and the Martin Marietta output has been

duplicated. No major problems are expected in meeting the compatibility requirements between Denver and Houston. The CalComp plotting hardware is identical at both locations.

Training courses have been offered to NASA and Martin Marietta personnel in Houston, instructing them in the operation of the NEWTIME and NEWCREW programs. The instruction book used in conjunction with these courses is included as Appendix B.

CHAPTER 3

COST-EFFECTIVENESS ANALYSIS

Approach to Cost-Effectiveness Analysis

The objective of this thesis was to develop cost-effectiveness methodology relevant to an analysis of computerization of the production of mission timelines for the NASA Skylab program. In order to accomplish this objective various cost-effectiveness techniques were developed and applied.

The reason for employing cost-effectiveness techniques rather than a straight engineering economy analysis of the alternatives in this thesis is much the same as the reason that cost-effectiveness techniques were originally developed in the Department of Defense (DOD). In attempting to evaluate many complex weapons and military systems in the DOD, it became apparent that the systems could not always be compared by using the strictly monetary values that would result from engineering economy studies. Therefore, when analyzing systems where their effectiveness was of some consequence, cost-effectiveness techniques were developed and employed to compare the systems. In the case of this thesis the effectiveness of the alternatives is important enough to warrant a cost-effectiveness analysis.

The first step employed to accomplish this task was to establish a set of parameters that define a means of comparatively analyzing the effectiveness of the competitive systems. These effectiveness parameters were then assigned subjective values designed to reflect their performance in

meeting established effectiveness goals. Weighting values were also assigned to performance parameters to reflect their contribution to the total systems performance. Eventually these values were used to calculate an expected value of effectiveness for each alternative.

Next, cost studies were performed to establish expected costs for each of the alternatives. An engineering economy analysis of the computerized method was conducted to assess the rate of return and the incremental rate of return on the system. A minimum attractive rate of return (MARR) was established as a method of determining whether the rate of return justifies the investment. A sensitivity analysis was conducted to determine the effect of an overrun in computer utilization estimates.

Eventually a ratio model was employed to determine a systems expected value of effectiveness for each alternative. From this a decision was made as to which alternative is more "cost-effective". Prior to the engineering economy analysis and the ratio model, a domain of feasibility study was conducted to determine if both alternatives are within cost and/or effectiveness thresholds. From this study a decision was made with respect to whether one or both alternatives were within resonable cost and utility boundaries.

In analyzing the alternatives in this study, the domain of feasibility technique was attempting to isolate one of the alternatives as having either excess cost or deficient

effectiveness when compared with its counterpart. By establishing a viable limit on minimum effectiveness and maximum costs, the domain of feasibility study provides graphical information about the alternatives and their relationship to each other. The feasibility study would be extremely beneficial in choosing between systems when faced with a large number of competitive alternatives by limiting any further studies to alternatives within the reasonable cost and effectiveness domain. In the case of the two alternatives in this thesis, it will be used to graphically depict the cost and effectiveness of each system relative to the other.

Next a homogeneity study was conducted on the parameters of time and possession to insure that both alternatives are feasible under these parameters. The homogeneity of time study examined each alternative to see if one has a much longer time of attainability than the other. If so steps were taken to penalize the one with the longer attainability time. The homogeneity of possession study analyzed both alternatives with respect to the skills necessary to operate each system. If one system was found to require skills that would significantly increase the cost of operation, steps to penalize that alternative were taken.

Following the homogeneity study of time and possession, a scale homogeneity analysis was conducted using a ratio model and an indifference curve mode. Homogeneity with respect to

scale permits the identification of systems that have the highest figure of merit. By combining monetary expected values with effectiveness expected values this figure of gain may be deduced. In the case of this study the respective figures of gain may be compared to appraise alternatives.

In order for most of the above mentioned techniques to be utilized, cost and effectiveness factors had to be developed to quantitize the necessary cost and effectiveness parameters. The methodology used to accomplish this is the objective of this thesis.

Cost-Effectiveness Analysis

The cost-effectiveness study of the competitive systems will be conducted in three parts. First, effectiveness parameters will be established that provide an objective way to compare the effectiveness of each alternative in meeting the final goal of timeline generation. Once established, these parameters will be assigned quantative values with the final goal being the establishment of an expected value of effectiveness for each alternative. Next, a cost analysis will be conducted on alternative II (computerization) along with a sensitivity analysis of the cost parameters. These values will then be compared to an established minimum attractive rate of return (MARR). An expected value of cost for each alternative will also be derived for use in the third and final step. Finally, using the cost and effectiveness expected values derived in steps one and two, a

ratio cost-effectiveness model will be used to establish the expected value of a cost-effectiveness figure of merit. This cost-effectiveness figure of merit will then be used as a means of establishing the cost-effectiveness superiority of one of the two alternatives.

EFFECTIVENESS MEASURES - The first step in the conduct of the cost-effectiveness analysis of the competitive alternatives outlined in this thesis will be to establish the parameters to be used in measuring and comparing the effectiveness of each system. Prior to the establishment of these parameters it will be helpful to discuss the decision processes that will be employed to assign expected values and weights to the effectiveness parameters. Since the assignment of these values is subjective it is necessary to remove as much of the arbitrary nature of their assignment as possible. If one person were to assign these values by himself, his assessment of their worth may be unduly biased due to his own personal opinion of the alternatives. His assignments may not accurately reflect the overall relative worth of each parameter when looked at from the larger systems overview.

A large amount of the biased subjective assignment of values inherent in the singular establishment of these values can be eliminated by discussing individual initial estimates with others knowledgeable on the system. In other words, all individuals familiar with the system can assign expected

values and weighting factors according to their own preferences. Once this has been done the group can compare their values and discuss any discrepancies. In this way any differences can hopefully be resolved through the discussion and values that seem reasonable to all parties can be established. This interplay of ideas should help those involved achieve a higher confidence level in their assignments by subjecting each of their criteria to inspection by others knowledgeable in the effort.

This type of approach was subconsciously used in establishing the values in this study. Over the period of time from the conception of timeline generation to implementation, the alternatives of manual and computerized generation were weighted and trade off studies conducted as to the relative merit of each alternative. Flight planners, NASA contract managers, astronauts and programmers were all actively involved with these studies and it is through these discussions, that have stretched over the period of a few years, that the values that will be assigned to effectiveness expected value and weighting were established. With the final objective of efficiently producing high quality mission timelines within schedule constraints as a goal, the following parameters represent the relevant effectiveness measures of a system to generate mission timelines.

- 1) FUNCTION: The function of each system will be analyzed as to its respective ability to meet the

requirements imposed in producing mission timelines. Requirements relative to the amount of information that must be depicted on the final output and the reaction or turn around time required to generate this output will be considered under this heading.

- 2) ACCURACY: Since the quality of the final output is important it will be necessary to compare each alternative on a qualitative basis. The accuracy of the data on the timeline frames will be considered under this heading as well as the consistency of output obtained.
- 3) ADAPTABILITY: Since the final objective of timeline generation is to eventually provide real time support during the Skylab mission, each alternative must be analyzed with respect to its ultimate adaptability to real time operation. It must also be readily adaptable to any changes that may be required due to trajectory changes or output format changes.
- 4) RELIABILITY: In premission and mission planning, timeline generation is a very time critical task. The ability of each system to perform when called upon must be considered and comparisons conducted to see which system is capable of meeting satisfactory reliability requirements.
- 5) AVAILABILITY: It will, of course, be important to consider the availability of each system. Time

required to gear-up and implement each alternative will be considered for each alternative.

The first three parameters - Function, Accuracy, and Adaptability - will be considered as performance parameters in agreement with Seiler's definition in "Introduction to System's Cost-Effectiveness".

Now that a workable set of effectiveness parameters has been established, a comparison of Alternative I (Manual timeline construction) with Alternative II (Computerized timeline construction) will be conducted within the framework of the aforementioned parameters. Quantitative values designed to measure the expected value of each alternative for each effectiveness parameter will be assigned at the conclusion of the discussion of each parameter.

The method of assignment of these expected value quantities will be to rate the competitive alternatives on a scale from 0.0 to 1.0 according to the following logic. A 0.0 expected value will represent unacceptable performance of the parameter and a 1.0 expected value will represent excellent performance. All performances falling between these two extremes will be assigned a value lying between 0.0 and 1.0 depending on their performance in satisfying the effectiveness criteria.

At the same time a weighting factor will be assigned to each of the first three parameters - function, accuracy and adaptability. This weighting factor will reflect the

importance of that one effectiveness parameter with respect to the entire system performance. These weighting factors will be in terms of % importance of that one parameter to the overall goal of producing mission timelines. The sum of the three weighting factors must therefore total 100%.

FUNCTION - The first effectiveness parameter requiring analysis is the function parameter. This parameter was established to address the problem of producing the mission timelines within the time allocated from trajectory initialization to required publication dates with the amount of information required to make the timelines a meaningful tool for flight planning. Alternative I, the manual method, is a very satisfactory method for implementation of timeline construction as long as the amount of information requiring depiction on the frame is limited in scope and complexity. The procedure of having a draftsman read time data from a computer printout and manually convert this data to distances on a linear time scale is a very tedious and time consuming process. As an example, the very early timeline frames, which consisted of only five sets of variables, took a draftsman about as long to complete as is currently required by the computer to complete a frame consisting from four to five times the number of variables required on the early frames. Therefore, it can be concluded that, as the amount of information required on the frame increases and the time allotted for frame generation remains the same the computer

generation method would hold a decided advantage over the manual method in function. With current requirements for reaction time and amount of information required, the functional effectiveness of the manual method would have to be rated somewhere between unsatisfactory and poor with an expected value of function $E(F)_i$ of .15. On the other hand, the functional effectiveness of the computer method should be in the range of good to excellent with an $E(F)_{ii}$ of .35.

The function of the alternatives is the most important effectiveness measure of the three established performance parameters contributing 50% to the total overall system performance. Therefore the weighting factor assigned to this parameter will be $W(F) = .50$.

ACCURACY - The next effectiveness parameter requiring analysis is the accuracy parameter. This parameter was established to compare the accuracy and consistency of output obtained by using each of the two alternatives. Accuracy in the manual method is a function of how well a draftsman can read time data from a computer printout, convert the time data to linear distances for time scale plotting and incorporate the data in the appropriate area on the timeline frame. Assuming that the draftsman is given a reasonable amount of time to process a reasonable amount of data, his accuracy in the task completion should fall in the good to excellent range. However, due to human error, the consistency of output may vary from frame to frame and

mission to mission and can therefore only be rated as from satisfactory to good.

The computerized method, on the other hand, should perform almost flawlessly in the area of accuracy and consistency. Computer output, once programs are running in batch mode, will produce identical outputs given identical inputs with the only exception being the possibility of erroneous input information.

Expected values for each of the alternatives can therefore be assigned as $E(A)_i = .75$ and $E(A)_{ii} = .95$. The accuracy effectiveness parameter contributes 30% of the weight to systems performance therefore $W(A) = 30\%$.

ADAPTABILITY - The adaptability of the system is important for two reasons. First, during the Skylab flights, real time or near real time support will be required in the area of mission timeline generation. The manual method of timeline generation does not lend itself to this type of near real time support for the reasons mentioned previously under performance. As the amount of data output increases the manual method becomes extremely time consuming and therefore undesirable for close real time support.

The speed with which the computerized method handles and plots comparable amounts of data makes it extremely adaptable to real time mission support. The time from receipt of input information to output production is good compared to the unsatisfactory time span associated with the manual approach.

Both methods would react comparably to changes in output requirements. The manual method would require the draftsman to construct his frames differently and incorporate different data while the computerized approach would require program modifications with neither method showing a decided advantage in this area. However, due to the computerized advantage in the area of real time support, an expected value of $E(D)_i = .50$ will be assigned to the manual method and $E(D)_{ii} = .75$ will be assigned to the computerized approach. The value of the adaptability of this system is not as important to the overall systems performance as are accuracy and function so a weighting factor of $W(D) = 20\%$ will be assigned.

RELIABILITY - Reliability, the next effectiveness parameter, is the measure of the dependability of each of the systems to perform when called upon. Assuming that both are capable of producing the desirable output within the time constraints imposed, both systems would perform in the good to excellent range of reliability. Computers experience down time and software problems and the manual method would be hampered by lost time and inefficiencies of draftsmen but all in all the reliability of alternatives I and II should be nearly equal. Therefore effectiveness values of $E(R)_i = E(R)_{ii} = .75$ will be assigned.

AVAILABILITY - Finally the availability of each of the alternatives must be weighed. Since a decision was made to pursue the computerized approach to mission timeline

generation, the manual method of generation has been abandoned. No effort has been made to maintain or expand upon the pool of draftsmen that may be required to manually construct mission timelines. It is doubtful that with the time remaining to the launch of Skylab that these manual skills could be satisfactorily developed and sufficient numbers of draftsmen hired and trained to adequately support timeline needs.

The computerized approach is currently operational and generation mission timelines in support of Skylab. Availability expected values of $E(V)_i = .30$ and $E(V)_{ii} = .95$ will be assigned to reflect the potential availability of each alternative.

It must be realized that the assignments of the expected values and weighting factors in the above effectiveness parameters is subjective in nature. In developing such a model, a criterion for the assignment of weights can be based on a measurement of the "parts" contribution to the performance of the "whole" system. It is at this stage that many undefendable assumptions may be created according to Seiler's "Introduction to System's Cost-Effectiveness".

Now that the expected values of the efficiency parameters and the weighting factors have been established, models do exist that measure the expected value of total system effectiveness $E(E)$ and one of these models will be employed in these analyses to establish $E(E)$ values for the

competitive alternatives in this study. The goal of comparing the methods of preparing timelines will be to establish a single expected value for each alternative and to use this single value to compare their relative merits.

The expected value of each system will be derived using the following equation as stated in Seiler's "Introduction to Cost-Effectiveness":

$$E(E) = [E(P)C_p] [E(V)C_v] [E(R)C_r] [E(S)C_s]$$

where $E(E)$ = expected effectiveness of system

$E(P)$ = expected performance of system

C_p = statistical confidence in performance

$E(V)$ = expected availability of system

C_v = statistical confidence in availability

$E(R)$ = expected value of reliability

C_r = statistical confidence in reliability

$E(S)$ = expected survivability of system

C_s = statistical confidence in survivability

It can be seen that the above technique establishes the system expected value by factoring the expected values of the parameters that comprise the system.

The survivability of the systems are of small consequence and are therefore eliminated from the above equation leaving:

$$E(E) = [E(P)C_p] [E(V)C_v] [E(R)C_r]$$

This equation assumes equal weights of the performance parameters. The equation modified to include weighting factors will appear as follows:

$$E(E) = \sum [W_i E(P_i) C(P_i)] [E(V) C_v] [E(R) C_r] / \sum W_i$$

The solution of this equation for each of the alternatives is shown in Appendix C and the results are as follow:

- 1) Expected value of Alternative I = $E(E)_i = .0656$
- 2) Expected value of Alternative II = $E(E)_{ii} = .4460$

This then completes the first step in the cost-effectiveness analysis of the computerized verses the manual generation of mission timelines. Expected values for each of the alternatives have been established for use in the ratio cost-effectiveness model in step three of the analysis. It may be helpful at this point to conduct a sensitivity analysis on the weighting factors used for the performance parameters. However, since the expected value of alternative II is greater by a factor of 7 than the expected value of alternative I, it is my feeling that simply varying any or all of the weighting factors for sensitivity purposes would not significantly alter the relative expected value results.

COSTS - We may now proceed to step two of the cost-effectiveness study which involves establishing expected values for the cost of each alternative as well as an engineering economy analysis to determine the rate of return for the computerized system. A sensitivity analysis will

also be conducted on the computerized approach to assess the effect of cost overruns of computer time.

As is the case in many projects dealing with aerospace programs cost is really not a restrictive factor. Generally the effectiveness of the system is what is critical and cost, as long as it is within reasonable limits, can be adjusted to obtain the effectiveness. In order to define baseline costing figures for this analysis \$90,000 per 3 month period will be established as an absolute ceiling of available funds to accomplish the task. Should costs exceed this \$90,000 figure costs would probably be deemed exorbitant and further effort discontinued. The cost quotient or cost expected value will then be expressed as a percent of this \$90,000 ceiling. We can now calculate the expected value of cost of the manual method.

This quantity is difficult to estimate due to the fact that a mission timeline of the current complexity has never been attempted manually. The best that can be accomplished is to attempt to extrapolate figures that are available for the effort required to complete the more simplified manual versions.

The early manual construction of the timeline required a total of 1 man month of effort for data preparation and 1.5 man months of data illustration as well as the 3 man months of flight planning assistance that are currently required for a mission. This is a total of 5.5 man months of effort to

construct the earlier simplified version of the timeline. The current version is approximately three times more complicated than the earlier version that required 5.5 man months of effort. We can assume that the flight planner support would remain constant at 3 man months, however, the data preparation and data illustration effort would increase in direct proportion to the complexity or from 2.5 man months to 3×2.5 or 7.5 man months making a total of approximately 10.5 man months of effort to complete the task.

10.5 man months @ \$4000 per man month = \$42,000.

Ratioing this \$42,000 on the maximum ceiling of \$90,000 we obtain the cost quotient.

$$C(Q)i = \$42,000 / \$90,000 = .467$$

Now that we have established an expected value for the cost of the manual technique, we can calculate the expected value for the cost of the computerized technique.

The calculations are as follows and are based on a typical 3 month period:

Approximate computer usage:

A) CDC 6400-6500 5 hrs.x \$550/hr = \$2750

The 5 hour figure was derived by assuming 1.5 hours run time to complete 3 Skylab NEWTIME runs and 1.0 hours run time to complete 3 Skylab NEWCREW runs.

This totals 2.5 hours and it is standard procedure in order to correct errors to execute each program in

this series twice thus 2 executions x 2.5 hours per execution = 5 hours.

B) CALCOMP Plotter 70 hours x \$40/hours = \$2800

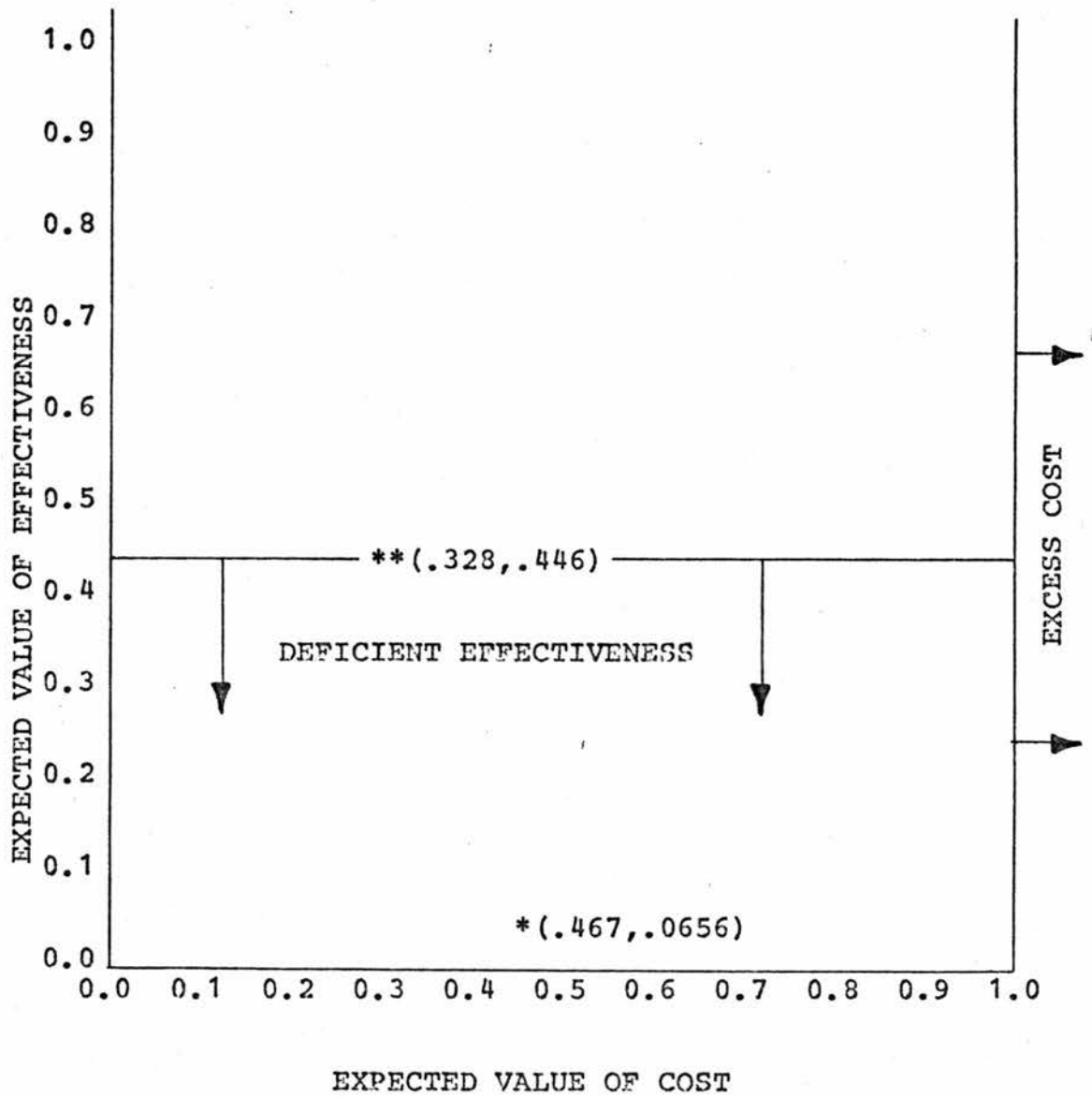
The 70 hours figure was derived by assuming 22.5 hours plot time to complete 3 Skylab NEWTIME plots and 12.5 hours to complete 3 Skylab NEWCREW runs.

This totals 35 hours and it is again standard procedure to run each plot twice to correct errors thus 2 plots x 35 hours per plot = 70 hours.

C) Manpower 6 man months x \$4000/man month = \$24,000

This 3 man month estimate is based on one programmer working full time on program modification and execution plus 3 flight planners working part time planning the mission and data coding the activities. In order to compensate for the part time status of the flight planners it was assumed that only men working full time could duplicate the effort of the 3 men working part time thus (1 flight planner + 1 programmer) x 3 months = 6 man months.

Materials and miscellaneous additional charges may occur during this period but the major portion of the cost of operation is covered in these three areas. Summarizing these costs it can be seen that the typical computer based cost of a 3 month flight planning effort is \$29,550.



- * ALTERNATIVE I - MANUAL METHOD
- ** ALTERNATIVE II - COMPUTERIZED METHOD

FIGURE 4 - DOMAIN OF FEASIBILITY CHART

Ratioing this figure to the \$90,000 ceiling we obtain the cost quotient or expected value of:

$$C(Q)_{ii} = \$29,550 / \$90,000 = .3280$$

for the computerized system.

Now that the cost and effectiveness parameters are defined it is possible to proceed with the domain of feasibility study.

The effectiveness parameters will be plotted on the ordinate of a graphical "domain of feasibility" block. The abscissa will contain cost parameters that can be applied to both alternatives.

Figure 4 shows the domain of feasibility block.

It can be concluded from the domain of feasibility study that Alternative I - the manual method - falls below the acceptable effectiveness minimum but is within the maximum cost limit. Its effectiveness, although deficient, is marginal and does not appear so low as to exclude further study. Alternative II - the computer method - falls well below cost maximums and above effectiveness minimums. From this initial rough study the computerized method would appear to be the more feasible of the two alternatives.

The maximum expected values for cost and the minimum expected value for effectiveness were established at points which would probably be considered reasonable by a person faced with deciding between these two systems.

The maximum cost figure used in the domain of feasibility study is the \$90,000 figure established for use in calculating the cost quotients (1.0 on the abscissa). The minimum effectiveness of .446 was obtained by rationalizing that you would want to obtain the maximum effectiveness that you could - in this case the effectiveness of computerization - and therefore would not be willing to accept anything below that maximum.

Even though, following the domain of feasibility study, Alternative I has been determined to be deficient in effectiveness, a decision to continue analyzing both alternatives seems reasonable. It is now appropriate to run homogeneity studies on time and possession. This is necessary because there are instances in which the alternatives may be heterogeneous with respect to one or more of these factors.

In studying the time homogeneity both alternatives must be looked at with the idea of penalizing an alternative that may have an unrealistically long time of obtainability. In analyzing the homogeneity of both alternatives with respect to time, a strange observation can be made concerning the alternatives. If this analysis had been performed during the period when the manual method was in use, the computerized approach would have had to have been penalized due to its unavailability at that time. However, the fact that a decision was made to abandon the manual method in favor of

the computerization, would now necessitate the penalization of the manual method. The penalization in accordance with this discussion appears as the lower expected value assigned to the manual method in the availability effectiveness parameter.

The homogeneity of possession study calls for a comparison of both alternatives to insure that the people who will be responsible for operating the systems possess the necessary skill levels to make the system function. Each of the alternatives in this study require specific skill levels to operate. Although the skill levels are vastly different - draftsmen verses programmers - they are both readily available and therefore the alternatives may be considered homogeneous with respect to possession.

This availability differs from the availability discussed as an effectiveness parameter in that this "skill level" availability refers only to skill levels required to complete the task. The "effectiveness" availability, on the other hand, contrasts the availability of the two systems - manual and computerized - and assigns expected values accordingly. The "skill level" availability is definitely a factor that was considered in "effectiveness" availability but only one of the factors. Such things as computer availability, software availability, etc., were as also factored into the consideration of "effectiveness" availability.

Table 1: Developmental and Operational Costs Associated with
Computerized Timeline Generation

Type of Cost:

1) Developmental (1 year basis)

a) Programming	13 MM
b) Test, check out and Data Conversion	2 MM
c) Computer Time (25 hrs. @ \$550/hr.)	\$13,750
d) Plotter Time (350 hrs. @ \$ 40/hr.)	<u>\$14,000</u>

Total 15 MM + \$17,750

2) Operational (Extrapolated From Domain of Feasibility 3 Month Study)

a) Computer Time (20 hrs. @ \$550/hr.)	\$11,000
b) Plotter Time (280 hrs. @ \$ 40/hr.)	\$11,200
c) Manpower	<u>24 MM</u>

Total 24 MM + \$22,200

MM = Man months

Engineering Economy Analysis

It is now possible to conduct an engineering economy analysis on what appears to be the superior of the two alternatives - Alternative II - the computerization of mission timeline generation. The first aspect of the cost analysis requiring study is the cost, to the company, to develop the techniques necessary to conduct computerized timeline generation. Costs associated with manpower needed in programming, test, checkout and data conversion must be established as well as the costs associated with computer time used to develop the programs. Table 1 reflects feasible values for developmental costs and includes information relative to operational costs that were derived in the domain of feasibility study.

Having established the necessary developmental and operational costs, a before taxes cash flow table may be constructed reflecting expected cash flow for the period of the next four years. Table 2 is the Cash Flow Before Taxes (CFBT) table.

Based on that data depicted in Table 1 and Table 2 and assuming, for tax analysis purposes, the computer hours are tax deductible while the man-hours are non deductible and taxable at a 50% rate, a cash Flow After Taxes (CFAT) table can be generated.

Table 2: Cash Flow Before Taxes

Year End	CFBT	
0 (NOW)	Development Costs:	
	Manpower: (15 MM @ \$4000/MM)	\$60,000
	Computer:	<u>\$17,750</u>
	Total	\$77,750

1	Received from NASA for manual method (33.5 MM @ \$4320/MM)**	\$144,500
	Cost of Manual Method (33.5 MM @ \$4000/MM)	<u>\$134,000</u>
	Total Profit	\$ 10,500

2	Received from NASA (33.5 MM @ \$4320/MM)	\$144,500
	Cost of Computer Time (20 hrs. @ \$550/hr.)	\$11,000
	Cost of Plotter (280 hrs. @ \$40/hr.)	\$11,200
	Manpower (24 MM @ \$4000/MM)	\$96,000
		<u>\$118,200</u>
	Total Profit	\$ 26,300

3 and 4	Same as 2	

** \$4320 calculated as \$4000 + 8% profit fee
MM = Man months

CFAT TABLE

Year	Tax	Taxable	Taxes	
End	CFBT	Deduction	Income	@ 50% CFAT
0	-77750	-17750	-60000	+30000 -47750
1	+10500	NONE	+10500	- 5250 + 5250
2	+26300	-22200	+ 4100	- 2050 +24250
3	+26300	-22200	+ 4100	- 2050 +24250
4	+26300	-22200	+ 4100	- 2050 +24250

Now we need to determine the Present Worth from the above table. To do this we can use the rate of return method for determining present worth (PW).

Try $i = 20\%$:

$$\begin{aligned}
 PW &= -47750 + 5250 (SPPW, 20\%, 1) + 24250 (USPW, 20\%, 3) (SPPW, 20\%, 1) \\
 &= -47750 + 5250 (.8333) + 24250 (2.106) (.8333) \\
 &= -47750 + 4370 + 42500 = -880
 \end{aligned}$$

Try $i = 10\%$:

$$\begin{aligned}
 PW &= -47750 + 5250 (SPPW, 10\%, 1) + 24250 (USPW, 10\%, 3) (SPPW, 10\%, 1) \\
 &= -47750 + 5250 (.9091) + 24250 (2.486) (.9091) \\
 &= -47750 + 4760 + 54900 = 11910
 \end{aligned}$$

Interpolation to the correct rate of return gives:

$$i = .10 + .10 (11910 / 12790) = .10 + .0935 = .1935 = 19.35\%$$

We can now see that the 19.32% rate of return is well above the 8% profit rate included in the calculations on Table 2. It is now possible to construct an incremental cash flow table to calculate what the incremental rate of return

Table 3: Cash Flow Before Taxes - 10% Computer Overrun

Year End	CFBT	
0 (NOW)	Development Costs:	
	Manpower: (15 MM @ \$4000/MM)	\$60,000
	Computer: (\$17,750 + 10%)	<u>\$19,525</u>
	Total	\$79,525
1	Received from NASA for manual method (33.5 MM @ \$4320/MM)**	\$144,500
	Cost of Manual Method (33.5 MM @ \$4000/MM)	<u>\$134,000</u>
	Total Profit	\$ 10,500
2	Received from NASA (33.5 MM @ \$4320/MM)	\$144,500
	Cost of Computer Time (20 hrs +10% @ \$550/hr)	\$12,100
	Cost of Plotter (280 hrs +10% @ \$40/hr)	\$12,320
	Manpower (24 MM @ \$4000/MM)	\$96,000
		<u>\$120,420</u>
	Total Profit	\$ 24,080
3 and 4	Same as 2	

** \$4320 calculated as \$4000 + 8% profit fee
MM = Man months

for computerization. The incremental cash flow table appears below:

CFAT TABLE					
Year		Tax	Taxable	Taxes	
End	CFBT	Deduction	Income	@ 50%	CFAT

0	-77750	-17750	-60000	+30000	-47750
1	0	NONE	0	0	0
2	+15800	-22200	- 6400	+ 3200	+19000
3	+15800	-22200	- 6400	+ 3200	+19000
4	+15800	-22200	- 6400	+ 3200	+19000

We can now calculate the incremental present worth and rate of return from the above table.

Try $i = 10\%$

$$\begin{aligned}
 PW &= -47750 + 19000 (\text{USPW}, 10\%, 3) (\text{SPPW}, 10\%, 1) \\
 &= -47750 + 19000 (2.486) (.9091) \\
 &= -47750 + 43000 = -4750
 \end{aligned}$$

Try $i = 5\%$

$$\begin{aligned}
 PW &= -47750 + 19000 (\text{USPW}, 5\%, 3) (\text{SPPW}, 5\%, 1) \\
 &= -47750 + 19000 (2.723) (.9524) \\
 &= -47750 + 49300 = +1550
 \end{aligned}$$

Interpolation to the correct rate of return gives:

$$i = .05 + .05(1550/6300) = .05 + .0123 = .0623 = 6.23\%$$

In analyzing the rate of return of 19.35% on the total return and 6.23% on the incremental return we can conclude that the after taxes rate of return on the total return is

substantially larger than the 8% negotiated rate of return. However, the incremental rate of return of 6.23% is below the 8% minimum attractive rate of return (MARR) but not enough to warrant concluding that the implementation of computerized timeline generation was necessarily a bad decision. We can therefore conclude that computerized timeline generation was an advantageous decision from a cost standpoint.

A sensitivity analysis of the alternatives at this point with respect to possible overruns in the areas of either manpower or computer time estimates will provide information as to the criticality of a variance from the cost estimates above. An analysis of the cases where estimates exceed by 10% the proposed manpower needed or where the estimates exceed by 10% the proposed computer time will show the effect of such overruns on the incremental rate of return. Since the incremental rate of return with no overruns is already below the negotiated 8% MARR we can not hope to achieve an acceptable rate of return with overruns in either area.

We will first analyze the case of a 10% overrun in computer hours estimated. Table 3 shows the cash flow before taxes of the computerized method with a 10% overrun in computer hours. Now from Table 3 we can construct the following incremental cash flow table reflecting a 10% computer usage overrun.

CFAT TABLE - 10% computer time overrun

Year		Tax	Taxable	Taxes	
End	CFBT	Deduction	Income	@ 50%	CFAT

0	-79525	-19525	-60000	+30000	-49525
1	0	NONE	0	0	0
2	+13580	-22200	- 8620	+ 4310	+17890
3	+13580	-22200	- 8620	+ 4310	+17890
4	+13580	-22200	- 8620	+ 4310	+17890

We can now calculate the present worth and rate of return from the above table.

Try $i = 3\%$

$$PW = -49525 + 17890 (USPW, 3\%, 3) (SPPW, 3\%, 1)$$

$$= -49525 + 17890 (2.829) (.9708)$$

$$= -49525 + 49100 = -425$$

Try $i = 2\%$

$$PW = -49525 + 17890 (USPW, 2\%, 3) (SPPW, 2\%, 1)$$

$$= -49525 + 17890 (2.884) (.9804)$$

$$= -49525 + 50600 = +1070$$

Interpolation to the correct rate of return gives:

$$i = .02 + .01(1070/1495) = .02 + .00715 = .0272 = 2.72\%$$

This shows that an overrun of 10% in computer time would reduce the incremental rate of return to a very unsatisfactory 2.72%. Comparing this to the negotiated 8%, it can be concluded that an overrun of this magnitude could

make the computerized approach unfavorable from a cost standpoint.

It can now be deduced, without the necessity of a sensitivity analysis on a 10% manpower overrun, that a manpower overrun of the magnitude of 10% would reduce the rate of return even further than the 2.72% caused by a computer overrun. This is true because of the manpower costs contributing about 4/5 of the total costs of the computerized method and therefore any increment to the manpower estimates would drastically reduce the rate of return below 2.72%. Experience with this system also shows that if cost overruns do occur they are more likely to occur on computer time rather than manpower and will generally be less than 5% in severity.

Therefore it can be concluded that, since the incremental cash flow analysis with no overruns produced a rate of return slightly below the MARR of 8%, then deviations from the values estimated in costing the systems would surely reduce that rate of return to an unacceptable level. This is verified by the sensitivity analysis of a 10% overrun in computer time reducing the rate of return to 2.72%.

The assumption of a MARR of 8% for this analysis is reasonable due to the method of NASA funding used to support this task. The cost per man month used in calculating manpower costs includes general and administrative costs as well as overhead charges. The MARR is derived as the 8%

profit figure added to this man month charge and therefore can be considered a viable MARR for this analysis.

System Expected Value

We may now proceed to the third and final step in the cost-effectiveness analysis of alternatives I and II.

The ratio cost-effectiveness model basically measures the value of a system in terms of the ratio of its effectiveness to its cost. In cases where a large number of multiple alternatives exist, graphical techniques may be necessary to distinguish between alternatives of nearly equal merit. In the case of only two possible alternatives the merit of each system may be calculated using the following equation from Seiler's "Introduction to System's Cost-Effectiveness".

$$E(M_i) = E(E_i) / E(C_i)$$

where: $E(M_i)$ = expected value of the cost-effectiveness figure of merit of the i th system.

$E(E_i)$ = expected value of the effectiveness of the i th system.

$E(C_i)$ = expected value of the total cost of the i th system.

Applying this technique to the two alternatives posed in this thesis, we arrive at the following relative values of merit.

Alternative I - Manual method

$$E(M_i) = E(E_i)/E(C_i) = .0656/.467 = .141$$

Alternative II - Computerized method

$$E(M_{ii}) = E(E_{ii})/E(C_{ii}) = .4460/.328 = 1.36$$

This ratio model approach points to Alternative II as the one with the highest system efficiency 1.36 compared to .141.

An indifference curve model is not applicable to this study for two reasons. First, this technique is more conducive to handling three or more than three combinations of systems. Second, this model assumes a constant marginal rate of substitution between Alternative I and Alternative II. This is not true, therefore the results of the ratio model analysis must be accepted as conclusive in determining scale homogeneity.

CHAPTER 4

CONCLUSIONS AND RECOMMENDATIONS

The ratio model analysis that has been performed on the two competitive systems seems to indicate that the computerized approach to timeline construction is the more cost-effective of the two methods. In the establishment of effectiveness expected values for each alternative, it was concluded that the computerization of timeline generation was by far a more effective means of accomplishing the goal of efficient timeline generation. It was also concluded by conducting a cost analysis of the two competitive alternatives that when comparing costs per period of time, computerization also proved to require a smaller cash expenditure than manual production. However, an engineering economy analysis of computerization showed that, although the rate of return on the investment required for computerization would yield a return of 19.35% which was acceptable when compared to an established 8% MARR, the incremental rate of return of 6.23% was below the 8% MARR.

A sensitivity analysis on Alternative II produced results showing that any appreciable cost overruns in either computer or manpower estimates would further act to reduce the rate of return below the acceptable 8% MARR.

As was mentioned earlier in the cost analysis, in the case of many aerospace programs, cost is not an overriding factor in the decision process as to whether a system is or is not implemented. Generally the effectiveness of the system in meeting established goals is the critical factor

when deciding on the ultimate acceptability of a system. This is the case in this project. The much higher effectiveness exhibited through computerization of timeline generation would in itself be enough to sway the decision process in favor of computerization. The fact that the incremental rate of return falls below the 8% MARR is discouraging but in itself is not sufficient to eliminate the alternative.

In conclusion, after applying standard cost-effectiveness techniques to two competitive alternatives of constructing Skylab mission timelines, it can be concluded that a computerized method of meeting the goal is the more cost effective of the two alternatives. An unsatisfactory incremental rate of return may, under different circumstances, cause a rejection of implementation. However, the effectiveness superiority of computerization is sufficient to override any disadvantages in engineering cost areas.

BIBLIOGRAPHY

Barish, Norman N., Economic Analysis for Engineering and Managerial Decision-Making, New York, New York, McGraw-Hill Book Company, 1962.

Brown, Craig M., Computer and Interactive Graphics as Applied to Mission Analysis, New York, New York, American Institute of Aeronautics and Astronautics, Jan., 1973.

Chemical Rubber Company Standard Mathematical Tables, Cleveland, Ohio, The Chemical Rubber Company, 1969.

Grant, Eugene L., and Ireson, W. Grant, Engineering Economy, New York, New York, Ronald Press Company, 1964.

Gushee, Charles H., Financial Compound Interest and Annuity Tables, Boston, Massachusetts, Financial Publishing Company, 1961.

Lake's Monthly Installment and Interest Tables, Denver, Colorado, A. V. Lake & Company, 1959.

Seiler, Karl, Introduction to System's Cost-Effectiveness, New York, New York, Wiley - Interscience, 1969.

APPENDICIES

APPENDIX A

DETAILED DISCUSSION OF PLOTTING PROGRAMS

This appendix will provide additional detailed information about the plotting programs NEWTIME and NEWCREW mentioned briefly in the main text. The NEWTIME program, which plots the trajectory dependent portion of the timeline frame, is dependent upon NASA generated magnetic tapes for input information.

The computer magnetic tapes are generated in a format that is suited for use as a printout if the information is dumped to the printer. This means that the data is preceded with header information designed to aid a person who is reading the tape dump in determining what each of the numerical parameters signify. This header information, although valuable to a person reading the tape dump, is extraneous information when the tape is used as input to a computer program. This extraneous information increases the amount of time the program runs and therefore increases the cost of execution. There are currently two such tapes generated by the Mission Planning and Development Group at the NASA L. B. Johnson Space Center in Houston, Texas. One tape contains information pertinent to State Vector and Ground Track parameters of the orbit. This tape gives information about the Skylab orbit for the entire mission at points ten minutes apart. Although there are many valuable parameters in this file the programs only require information

relative to the Ground Elapsed Time (GET) in Days, Hours, Minutes and Seconds and in the geodetic latitude of the orbit at each discrete time point.

The second tape contains seven files in the following order: STDN Tracking Network, Sun Rise/Sun Set Tables, Revolution Counter Time History, Ascending Node Time History, Flight Dynamics Officer (FDO) Detailed Maneuver Table, Beta Angle Time History and a Moon Rise/Set Table. Of the seven files, six are necessary for the operation of the NEWTIME flight planning program. The only file not required is the FDO Detailed Maneuver Table. Of the six required files, only certain bits of data are necessary for the execution of the NEWTIME program.

In order to extract the pertinent data from each of these tapes, two reduction programs were developed that when executed with these two NASA tapes as input, pull necessary data from the tapes and reformat it so that it is in the proper form to be used as input to the NEWTIME program.

Once the reduction programs have reformatted the data, the program known as NEWTIME that plots the graphical underlay is ready to be executed. The output of this program is a magnetic tape (or in the case of a 56 day mission 2 magnetic tapes) that interface with a CalComp plotter to produce graphical outputs of the type shown in Figure 2 in the main text.

NEWTIME is actually the name of the main program that calls upon various subroutines to accomplish the plotting of the mission parameters in their proper location on the timeline. The main program initializes parameters and increments them accordingly during the program as well as calling the necessary subroutines. The NEWTIME program has 12 subroutines that are responsible for various phases of the plotting during the program. A brief discussion of each of the subroutines and what they accomplish will follow in the order that they are called from the main program. (See Appendix D for a printout of the NEWTIME program including subroutines.)

The first subroutine called during the program is subroutine FRAM1. This subroutine calls the necessary CalComp plotting subroutines to construct the outline of the frame that will contain the information as well as dividing up the frame internally into suitable blocks with tic marks that most effectively present the data. At one point in the development of the programs the possibility of having preprinted forms prepared with the basic outline of the frame printed on them was considered. However the necessity to react rapidly to changes in format necessitated the inclusion of this subroutine to construct the frame.

Next a logic test is performed in the main program to determine from input data if the mission to be plotted is conducted during Central Standard Time or Central Daylight

Savings Time. This fact effects the Greenwich Mean Time (GMT) scale as it appears on the frame. If the mission occurs during Daylight Savings Time the subroutine DAYSAV is called and the GMT scale is initiated with 11 GMT on the left most time hack. If the mission occurs during Standard Time the subroutine STAND is called and the GMT scale is initiated with 12 GMT on the left most time hack. This subroutine also enters the day of the year breakpoint and the mission day breakpoint on the frame. These two points, which appear above the GMT time scale, are plotted as vertical lines with numbers on each side of the line. These lines show the times when the day of the year and the mission day change on the frame. These points are helpful in determining exactly in which day an occurrence takes place.

After the GMT time scale is entered, the subroutine HEADER enters the alpha-numeric titles in the proper places on the frame. Again it was found to be more versatile to maintain the capability to change or move header information within the frame.

At this point in the program all of the constant data has been entered on the frame and the variable information that was reformatted from the NASA magnetic tapes and initialized in the main program is entered onto the frame in the proper time position. This segment of the program is where the computerization of this technique has proven most beneficial. The data from the NASA tapes reflect the occurrence of

various events such as times of sun sets or revolutions. These times are in the form of Ground Elapsed Times (GET's) measured from the launch of Skylab I. In other words, at the time of the launch of Skylab I, the GET is 0 Days, 0 Hours, 0 Minutes and 0 Seconds. Upon the launch of Skylab I a clock begins running and continues to run throughout all four Skylab missions measuring the number of days, hours, minutes and seconds since the launch of Skylab I.

The next subroutine is called DATMON. This subroutine enters the variable header information under the appropriate titles across the top of the frame. The first bit of information entered falls under the heading of GET. Unlike the GET referenced to the launch of Skylab I used on the NASA tapes, this GET is reset at the launch of each separate Skylab mission. This GET is expressed in terms of Days:Hours:Minutes and at the launch of each separate mission it is 0 Days: 0 Hours: 0 Minutes. The span of GET's on the top of the frame is always 24 hours showing the GET at the beginning and end of the frame.

Next this subroutine enters the mission day and day of the year. The mission day signifies the whole number of days that have elapsed from the launch of the current mission. If the launch occurs at 10:59 GMT on a Daylight Savings Time mission, mission day one in that case would only be one minute long. Every successive mission day is 24 hours in duration. The day of the year is just as it implies, January

first of every year is day of the year one, February first is day of the year thirty two etc. The next entry is the day of the week and the calendar date in Houston, Texas that corresponds to the day of the year.

The next three entries made by the DATMON subroutine are constant entries for each mission. The mission entry signifies the mission that is planned on the frame. SL-2 is Skylab II, SL-3 is Skylab III and SL-4 is Skylab IV. The edition denotes various phases of publications to differentiate from one phase of planning to the next. Finally the publication date is just what it implies, the date the documentation is released for usage at NASA.

The final function of the DATMON subroutine is to calculate the number of days that have elapsed since the last new moon prior to launch. Once this is known various trigonometric functions calculate points that depict a graphical representation of the phase of the moon as it appears in Houston, Texas on the Houston date. This is plotted in the moon phase sector of the frame.

The following subroutines convert the variable time data into distances that are plotted in the proper position on the timeline frame. The following technique is common throughout the remaining subroutines to convert time data to distances and plot it in the proper position on the timeline frame.

Files exist which contain various parameters related to orbital occurrences that happen during the mission. These

occurrences are quantitized such that the occurrence is described according to the GET that it occurs. This GET is the GET referenced to the launch of Skylab I. An example of this is the acquisition and loss of signal from a tracking station which would appear as follows on the tracking station file:

```
TEX      22 22 48 28.8      4. 34.4
```

Translating the data into understandable terms reveals that the tracking station Texas - abbreviated as TEX - acquires a signal from the spacecraft at 21 Days, 22 Hours, 48 Minutes and 22.8 Seconds and remains in contact for 4 Minutes and 34.4 Seconds.

The seven parameters in the tracking station file are read as the following seven variables:

```
STATION  = TEX
IDAYACQ  = 21
IHORACQ  = 22
IMINACQ  = 48
SECDACQ  = 28.8
IDELMIN  = 4
SECDELT  = 34.4
```

The integer values if IDAYACQ, IHORACQ, IMINACQ and IDELMIN are then floated or converted to real numbers for calculating purposes. The real number counterparts are:

```
DAYACQ   = FLOAT(IDAYACQ)
HOURACQ  = FLOAT(IHORACQ)
```

```
ACQMIN    = FLOAT(IMINACQ)
```

```
DELMIN    = FLOAT(IDELMIN)
```

Next everything is converted to minutes and expressed as a start and stop time for the occurrence. The acquisition of the station is referred to as STASTAT and is calculated as follows:

```
STASTAT = (DAYACQ*1440.)+(HOURACQ*60.)+ACQMIN+(SECDACQ/60.)
```

The time of the loss of the station is derived by adding the delta minutes of station coverage to the start time. This variable is STASTOP and calculated as follows:

```
STASTOP = STASTAT+DELMIN+(SECDELT/60.)
```

Now that the GET start and stop times are available for that one particular variable, logic checks are made to see if this variable falls within the 24 hour time period being considered. If it does the times are converted to distances by ratioing times and multiplying by a constant factor that converts time to distance in inches and plotted on the frame. If the occurrence does not fall within the 24 hour period being considered the program continues searching for parameters that do or proceeds to a different logic loop and searches different parameters.

The first variable parameter subroutine that is processed in this manner is BETANGL. This subroutine reads a file that contains the GET in Days, Hours, Minutes and Seconds of the time the Beta Angle reading is given. The Beta Angle is used to describe the angle between the orbit plane and the earth

to sun line and is necessary for planning certain experiments on the mission. Using the time and an angle which is read from the file, a small down arrow is plotted along the top of the GMT time scale to denote the exact time the beta angle is given. The value of the angle is then entered under the Beta Angle heading along the top of the frame.

The next time dependent subroutine to be processed is REVOLTN. This subroutine also reads time parameters and a revolution number to describe the occurrence. The revolutions depicted in the plotted output of this subroutine are defined as the time the spacecraft passes a constant longitude - 80 degrees West longitude. Everytime this occurs a new revolution has begun and a vertical line is plotted in the proper position to the right of the REV header and the number of that revolution is entered to the right of the line. The revolution numbers are initialized with the launch of Skylab I and continue incrementing throughout the entire four missions.

The subroutine DANIT is the next to be called. It functions by reading the GET of an orbital sun set and the delta time that the sun remains set. From this information it derives the start and stop time of orbital sun sets. This information is then converted to distances and plotted with dark areas denoting darkness or night time in the spacecraft and the light areas denoting daytime in the spacecraft. The numbers entered in the daylight areas aid planners of Apollo

Telescope Mount (ATM) experiments by differentiating between daylight cycles.

A similiar subroutine, MOONSET, functions by employing the same techniques of reading times of orbital moon set and a delta time of being set to plot light and dark areas in the MOON UP/DN area of the frame. In this case the light areas represent times when the moon is visible - assuming windows at the necessary viewing angle - to the orbiting spacecraft. The dark areas represent times when the moon is occulted by the earth and therefore not visible to the spacecraft.

The next subroutine - MSFNCV - reads a file containing information about times of coverage of the various tracking stations in the space flight network. The GET's of station acquisition and a delta time measured to the time of station loss are read from this file as well as a three letter designator that defines the tracking station being analyzed. There are 13 possible tracking stations that collectively form the Spaceflight Tracking and Data Network (STDN). They are, along with their three letter designators, as follows:

MIL - Merritt Island

BDA - Burmuda

NFD - Newfoundland

MAD - Madrid, Spain

CRO - Carnarvon, Australia

HSK - Honeysuckle Creek, Australia

GDS - Goldstone

TEX - Texas

CYI - Canary Island

HAW - Hawaii

ACN - Ascension Island

GWM - Guam

VAN - Vanguard Ship

When a station acquisition occurs within the 24 hour period being processed, a line is drawn from the point of station acquisition to the point of station loss and the three letter designator is entered to the right of the line. Passes of less than 2 minutes in duration are ignored due to the overlapping effect that occurs when all passes are plotted.

The final two subroutines in the NEWTIME program - ORBUN and WRDLMAP - complete the earth trace with South Atlantic Anomaly portion of the frame. This entry on the frame depicts the ground track of the spacecraft as it sweeps across the earth. The ground track is constructed from data read relative to the GET every ten minutes and the latitude and longitude that the spacecraft is passing at that time. Once the ground track is completed the subroutine WRDLMAP underlays an outline of the continents and the area of the South Atlantic Anomaly beneath the orbit trace so that at any chosen time the position of the spacecraft above the earth is known. Upon the completion of these final two subroutines the NEWCREW program, which plots crewman activities, is

executed. (See Appendix E for a printout of the NEWCREW program.)

The NEWCREW program has no subroutines, however, the main body is divided into two main computational loops. The first logic section processes data relative to the activities that the three crewmen perform during the mission. The second logic section processes data relative to the momentum dumps, scientific airlocks and unattended experiments. The technique used by the flight planners is as follows:

First the activities of each crewman for one 24 hour period are planned. The first step in the procedure used to accomplish this is to assign a numerical code to each possible activity that may be performed during the mission. A printout is available to the planner that correlates each activity with its associated numerical code. The flight planner, using his final penciled version as a guide, enters the code for the activity as well as a start and stop GMT in hours and minutes and a number that defines the crewman that will perform the task. He continues doing this until all of the activities for all three crewmen are planned for one 24 hour period. Now the flight planner is ready to enter the second logic loop that plans inhibits to momentum dumps, scientific airlock experiments and unattended experiments for the same 24 hour period.

Certain experiments require stabilization of the spacecraft while the experiment is being performed. For

example, an experiment designed to photograph a certain starfield may require a long camera exposure time to obtain the necessary photograph. Obviously motion during this period would destroy the possibility of performing such an experiment. There is a maneuver that must be performed periodically during the Skylab missions known as a gyro momentum dump. This maneuver requires that the spacecraft be rotated to accomplish the dump. For previously stated reasons it may, at times, be necessary to retard or inhibit this momentum dump. If this is the case, the flight planner must note this on his flight plan in the space denoted Control Momentum Gyro Desaturations (CMG DESAT). To accomplish this the flight planner has a code which he assigns and specifies the start and stop times of the inhibit. An inhibit momentum dump (IMD) as entered above the scheduled desaturation time to show that the maneuvers scheduled for that time must be inhibited.

Additionally, the flight planner must also note which experiments may be residing in the scientific airlocks at various points throughout the mission. There are two such airlocks, one pointing directly at the sun known as the Solar Airlock (SOL) and the other pointing 180 degrees away from the sun known as the Anti-Solar Airlock (A-SOL). To accomplish this there are four digit codes assigned to airlock experiments. The flight planner prefixes the appropriate code with either a 0 for a solar experiment or a

1 for an anti-solar experiment as well as the start and stop times in GMT hours and minutes that defines the period of time each experiment is within the airlock. Each period must be treated as a discrete 24 hour period and if the experiment carries over to another 24 hour period the flight planner must take this into account.

Finally, certain experiments must operate without crew intervention during the mission. These unattended experiments must also be tracked and entries made in the Unattended Experiments section of the timeline. Again a code is assigned to describe the activity and a start and stop GMT assigned to define the period of time it is running without supervision.

At this point a termination code is assigned to key the program to the fact that a day has been completed and the flight planner moves on to planning the succeeding days activities.

When a complete data file has been generated, it is executed with the NEWCREW program and a tape is generated that interfaces with the CalComp plotter. When plotted with the NEWTIME plot as an underlay the final resulting plot appears as in Figure 3 in the main text.

This completes the discussion of the plotting programs necessary to construct a complete mission timeline. Now that the contents of the programs have been considered, a brief

look at some techniques that were contemplated as a means of optimizing the plotting will be discussed.

The feasibility of combining certain aspects of the two alternatives - manual and computerized timeline generation - and arriving at a compromise system is a possibility. For example, a draftsman could construct the frame with the computer being used to enter the variable data. This alternative has been considered but, due to the criticality of a rapid response time, this possibility has been discounted. Since any combinations of manual and computerized systems could only have an adverse effect on response time, a mixture of the two systems could not be considered.

There are several ways that could be suggested to speed up the response time of the current computerized system beyond what currently exists. The programs, as they are written, are nearly optimal as far as the time it takes for them to execute. The bottleneck that currently exists in the system is the plotter. Once a tape is generated, it requires 12 hours to complete the plotting of a 56 day mission. This is due to the limitations imposed by the mechanical CalComp plotter. Current design restricts pen movements to a maximum of 8 inches/second. There is no way, other than optimizing pen movements within the program which has already been done, to speed up this plotting time on the current hardware.

A new CalComp plotter will soon be available on the market. This plotter will increase the maximum speed of pen movement from 8 inches/second to 50 inches/second and decrease plotting time by approximately 90%. This would mean that the current 12 hours of plotting could be completed in 2 to 3 hours.

Another alternative available is to utilize an electronic plotting tool - an FR-80 - to generate the plots. The tapes that are used with the CalComp plotter are compatible with the FR-80. The FR-80 flashes an electronic picture, according to the tape commands, on a cathode ray tube. A photograph of the tube is then taken producing a permanent hardcopy of the picture on the screen. There are size restrictions associated with using the FR-80 and certain program modifications would be necessary to make the current programs compatible. These program modifications would require development time but, once completed, would result in the least amount of time spent in the plotting loop.

This electronic process of generating timelines is the only one that is adaptable to a strictly real time application. In real time operation it would be necessary to obtain near instantaneous response to any sudden changes in orbital parameters such as a change in launch time, etc. A dedicated computer is necessary for this type of operation. This is a very expensive proposition, and it is doubtful that the benefits of such a venture would warrant the cost. The

less expensive near-real time operation, employing the CalComp plotter or FR-80, would probably be sufficient to meet the requirements that will exist when Skylab becomes operational.

APPENDIX B

INPUT & OPERATING INSTRUCTIONS
FOR FLIGHT PLANNING PROGRAMS
NEWTIME AND NEWCREW AND ASSOCIATED
SUPPORT PROGRAMS

June, 1973

Craig Matthew Brown

Martin Marietta Corporation

Introduction

The two programs outlined in this instruction manual are programs used by mission planners as graphic tools to display Skylab trajectory parameters and time dependent crew and mission functions. There are two categories of programs grouped as to the function they perform.

The program that constructs the timeline frame and enters the orbit dependent parameters such as GET's, GMT's, Beta Angles, Houston date, Revolution numbers, Day/night cycles, Network coverage, Moon phase and the earth trace with SAA is the NEWTIME program. This program calls all of the necessary plotting routines in the appropriate order to generate the completed timeline frame. This program requires two support programs that provide the orbit variable and earth continent coordinate data to the program. These support programs are MPADCM and WRLLDCM. The order of execution and purpose of these programs will be covered in Section A - NEWTIME series of programs.

The second group of programs has as its main program NEWCREW which graphically overlays the crewman activities performed by each crewman as well as experiment information onto the timeline frame generated by NEWTIME. The experiment information entered is pertinent to such things as times of IMD's, experiments in the SAL and times in the SAL and unattended experiments and times they are running. NEWCREW requires one computer support program as a data base

generator - CREWCM. These programs, their order of execution and their purpose will be discussed in Section B - NEWCREW series of programs.

One additional program exists as an aid to the flight planner. It gives him a current listing of what activities are available for him to use in planning a mission and the codes that are associated with these activities. This program is CAFDMP and will be discussed in Section B along with the NEWCREW program.

Section A - NEWTIME Series of Programs (NEWTIME, MPADCM and WRLDCM)

1. MPADCM - This program reads trajectory information from the MPAD trajectory tape and creates common files that contain this information. In turn these common files are accessed by the NEWTIME plotting program which uses this information to plot trajectory parameters in the correct time referenced position on the timeline frame. This program also provides a printed output of the information that is on the MPAD trajectory tape and may be used to check the tape for formatting and information content. Common files, once generated, will usually remain on the computer system for the "entire day" in which they are generated. In other words once MPADCM is run, NEWTIME may be run any number of times during the day without necessitating a rerun of MPADCM.

However if a day elapses between runs it will be necessary to rerun MPADCM prior to running NEWTIME so that the common files will be available for the program. Occasionally the operation of the system will require the dropping of common files during the normal daily operation of the system. It is wise to check with the operator if there is some doubt as to whether the files are on the system or not.

The common files generated by MPADCM and their content are as follows:

- a) CARASCN - GET as ascending node information
- b) CARMSEFN - MSFN tracking station information
- c) CARDAN1 - Lighting, day/night information
- d) CARBETA - Beta angle information
- e) CAREPHM - Latitude and longitude points
- f) CARREV1 - Revolution numbers and associated times

The only input required for the MPADCM program is a tape number that defines the MPAD trajectory tape. That tape number goes on the second card of the program in columns 20-24. This tape number will be assigned by the operator when given the tape and will be the same for every successive use of that tape.

2. WRLDCM - This program creates a common file consisting of latitude and longitude points that define the various continents of the earth. The points exist as data card input to the program.

There is no additional input required to the program and no output, other than a day file, is generated. The common file generated by this program is WRLDCM AND IS SUBJECT TO THE SAME RULES AS THE COMMON FILES GENERATED BY MPADCM.

3. NEWTIME - This program is the main program in this group. Once the necessary common-files are generated by MPADCM and WRLDCM this program draws upon that data and other input information to correctly construct the timeline frames.

The input requirements to this program are relatively simple consisting of only two input cards containing a total of 18 variables.

- a) Input card 1 - Contains 11 variables. A brief discussion of each variable, how it is deduced and where it is entered on the card follows:

- 1) Variable 1 (TIMFRM) (Columns 1 - 10,A10) - This variable tells the program whether the mission being run is a Standard time or a Daylight Savings time mission. If the mission occurs during Daylight Savings time, enter DAY in columns 1 - 3. If the mission occurs during Standard time, enter STD in columns 1 - 3.
- 2) Variable 2 (EDITN) (Columns 11 - 20,A10) - Enter any ten alpha-numeric characters to

describe the edition of the timeline being published. The information must be left justified.

- 3) Variable 3 (GEDAST) (Columns 21 - 25,F5.0) -
This variable initializes the GET days at the beginning of the Mission day 1 timeline frame referenced to the launch of that mission. In most cases this will be 0.0.
- 4) Variable 4 (GEHRST) (Columns 26 - 30,F5.0) -
This variable initializes the GET hours at the beginning of the Mission day 1 timeline frame referenced to the launch of that mission.
- 5) Variable 5 (GEMNST) (Columns 31 - 35,F5.1) -
This variable initializes the GET minutes at the beginning of the Mission day 1 timeline frame referenced to the launch of that mission.
- 6) Variable 6 (GEDASP) (Columns 36 - 40,F5.0) -
Same as variable 3 except for the end of mission day 1 frame (24 hours later than variable 3).
- 7) Variable 7 (GEHRSP) (Columns 41 - 45,F5.0) -
Same as variable 4 except for the end of mission day 1 frame (24 hours later).

Table 1B - TABLE OF DAYS BETWEEN TWO DATES

	January	February	March	April	May	June	July	August	September	October	November	December
1.	1	32	60	91	121	152	182	213	244	274	305	335
2.	2	33	61	92	122	153	183	214	245	275	306	336
3.	3	34	62	93	123	154	184	215	246	276	307	337
4.	4	35	63	94	124	155	185	216	247	277	308	338
5.	5	36	64	95	125	156	186	217	248	278	309	339
6.	6	37	65	96	126	157	187	218	249	279	310	340
7.	7	38	66	97	127	158	188	219	250	280	311	341
8.	8	39	67	98	128	159	189	220	251	281	312	342
9.	9	40	68	99	129	160	190	221	252	282	313	343
10.	10	41	69	100	130	161	191	222	253	283	314	344
11.	11	42	70	101	131	162	192	223	254	284	315	345
12.	12	43	71	102	132	163	193	224	255	285	316	346
13.	13	44	72	103	133	164	194	225	256	286	317	347
14.	14	45	73	104	134	165	195	226	257	287	318	348
15.	15	46	74	105	135	166	196	227	258	288	319	349
16.	16	47	75	106	136	167	197	228	259	289	320	350
17.	17	48	76	107	137	168	198	229	260	290	321	351
18.	18	49	77	108	138	169	199	230	261	291	322	352
19.	19	50	78	109	139	170	200	231	262	292	323	353
20.	20	51	79	110	140	171	201	232	263	293	324	354
21.	21	52	80	111	141	172	202	233	264	294	325	355
22.	22	53	81	112	142	173	203	234	265	295	326	356
23.	23	54	82	113	143	174	204	235	266	296	327	357
24.	24	55	83	114	144	175	205	236	267	297	328	358
25.	25	56	84	115	145	176	206	237	268	298	329	359
26.	26	57	85	116	146	177	207	238	269	299	330	360
27.	27	58	86	117	147	178	208	239	270	300	331	361
28.	28	59	87	118	148	179	209	240	271	301	332	362
29.	29	---	88	119	149	180	210	241	272	302	333	363
30.	30	---	89	120	150	181	211	242	273	303	334	364
31.	31	---	90	---	151	---	212	243	---	304	---	365

FOR LEAP YEAR, ONE DAY MUST BE ADDED TO
NUMBER OF DAYS AFTER FEBRUARY 28

- 8) Variable 8 (GEMNSP) (Columns 46 - 50,F5.1) - Same as variable 5 except for the end of mission day 1 frame (24 hours later).
 - 9) Variable 9 (MD) (Columns 51 - 55,I5) - This is the mission day that is depicted on the first timeline frame plotted. This number is right justified and will usually be 00001.
 - 10) Variable 10 (ID0Y) (Columns 56 - 60,I5) - This is the day of the year represented by the start of the first timeline frame. Table 1B will aid in determining this number. The number is right justified and therefore day 121 must be entered as 00121.
 - 11) Variable 11 (MISSION) (Columns 61 - 70,A10) - This is any group of ten or less alphanumeric characters, left justified, used to describe the mission being processed. Usually this will be either SL-2, SL-3, or SL-4.
- b) Input card 2 - Contains 7 variables. A brief discussion of each variable, how it is deduced and where it is entered on the card follows:
- 1) Variable 12 (ANEWMN) (Columns 1 -10,F10.2) - This is the day of the year on which the last newmoon occurred prior to the launch of the

Table 2B - DATES OF NEW MOONS IN TERMS OF DAY OF THE YEAR

<u>YEAR</u>	<u>DAY OF YEAR</u>
1973	4.13471
1973	33.74721
1973	63.40554
1973	92.96804
1973	122.52221
1973	152.09721
1973	181.75137
1973	211.45137
1973	241.05554
1973	270.52637
1973	299.92637
1973	329.30137
1974	-6.22363
1974	23.30971
1974	52.83054
1974	82.24721
1974	111.68471
1974	141.16387
1974	170.77637
1974	200.51804
1974	230.21804
1974	289.37221
1974	318.86387
1974	348.42221

mission being processed. Table 2B will give this information.

- 2) Variable 13 (GETREF1) (Columns 11 - 20,F10.2)
- This is the GET in minutes at the beginning of the first timeline frame being processed referenced to the launch of SL-1.
- 3) Variables 14 & 15 (PUBDAT1, PUBDAT2) (Columns 21 - 40,2A10) - These two variables are used to define the publication date of the timeline frames being processed. The publication date may be any 20 or less, left justified, alpha-numeric characters that give the date.
- 4) Variable 16 (YEAR) (Columns 41 - 50,F10.0) - This is the year of the launch date of the Skylab mission being processed.
- ##5) Variable 17 (LOWDA) (Columns 51 - 55,I5) - This is the first mission day that you wish to output. Enter as a right justified five digit number. For a normal mission this number will be 00001.
- ##6) Variable 18 (IHIDA) (Columns 56 - 60,I5) - This is the last mission day that you wish to output. Enter as a right justified 5 digit number. For a normal SL-2 mission the number

will be 00030. For a normal SL-3 or SL-4 mission the number will be 00060.

Examples of how to use variables 17 and 18 to run single day or series of days missions:

- 1) To run a single day timeline, say mission day nine, make LOWDA = 9 and IHIDA = 10.
- 2) To run a consecutive series of days, say mission days ten through thirteen, make LOWDA = 10 and IHIDA = 14.

Section B - NEWCREW Series of Programs (CREWCM and NEWCREW)

1. CREWCM - This program creates a common file data base of crewman activity codes and their corresponding activities. Like MPADCM and WRLDCM it is subject to the constraints of any common file. There is no input required in addition to the existing input cards.
2. NEWCREW - This program is the main program in this group. Once CREWCM generates the necessary common file this program searches the common file per input codes and times and overlays the crewman and experiment activities on the timeline frame generated by NEWTIME. The input cards to this program are as follows:
 - 1) Card 1 - The first input card is a card with either a zero or a one on column one. This card tells the program whether a daylight savings time

mission (designated by a zero in column one) or a standard time mission (designated by a one in column one) is being processed.

- 2) Cards 2 through XXX1 - This is a group of any number of cards used to define the activities of the various crewman at various phases "within" a mission day starting normally with mission day 1.

The contents of the cards per column are as follows:

- a) Column 1 - blank
- b) Columns 2 - 5 - A four digit code that describes the activity to be performed.
- c) Columns 6 - 10 - This entry may be arbitrarily made. If made it is the day of the year of the activity being coded. The entry can be left blank because it is ignored in the program.
- d) Columns 11 - 12 - Two digits right justified that tells the hour when the activity should START.
- e) Column 13 - blank
- f) Columns 14 - 15 - Two digits right justified that tell the minutes when the activity should START.
- g) Column 16 - blank

- h) Columns 17 - 18 - Two digit right justified that tell the hour when the activity should STOP.
 - i) Column 19 - blank
 - j) Columns 20 - 21 - Two digits right justified that tell the minutes when the activity should STOP.
 - k) Column 22 - blank
 - l) Column 23 - The crewman that will perform the activity. 1 = CDR, 2 = SPT and 3 = PLT.
 - m) Column 24 - blank
 - n) Columns 25 - 26 - If the four digit code in columns 2 - 5 is an EREP pass code, ie. 6720, the EREP pass number goes in these columns. The number must be two digits right justified.
- 3) Cards XXX1+1 through YYY1 - This is a group of any number of cards that define the start and stop times hr. and min. of IMD's, SAL experiments or unattended experiments.
- The content of the cards per column is as follows:
- a) Column 1
 - 1) blank followed by 8000 code is IMD code.

- 2) zero followed by any valid code greater than 8000 and less than 9999 indicates a SOL code.
 - 3) one followed by any valid code greater than 8000 and less than 9999 indicates an A-SOL code.
 - 4) two followed by an valid code greater than 8000 and less than 9999 indicates an unattended experiment code.
- b) Columns 2 - 5 - any code greater than or equal to 8000 and less than 9999.
 - c) Columns 6 - 10 - blank
 - d) Columns 11 - 21 - Same START and STOP indicators as Columns 11 - 21 on cards 2 through XXX1.
 - e) Columns 22 - 80 - blank
- 4) Card YYY1+1 - This card is a card with all 99999999's punched on it that signifies the end of one days planning. This card is followed by as many additional sets of cards needed to complete the mission.

NOTE: CARDS FROM GROUP 2 THROUGH XXX1 MAY NOT BE INTERSPERSED WITH CARDS FROM GROUP XXX1+1 THROUGH YYY1.....

3. CAFDMP - This program is an auxillary program that may be run with the crewman activities file as input data. The output of this program is a listing of the current crewman codes and what activity they correspond to.

The following is the format in which the crew activities data base is generated:

a) Crewman Activities Card Format

- 1) Columns 1 - 4 (Format I4) - Four digit code that represents the activity
- 2) Column 5 (Format I1) - Number of lines of information the code will use (Maximum 5 lines)
- 3) Column 6 (Format I1) - Angle at which the information will be written. A zero will cause the information to be written horizontally on the timeline frame and a one will produce vertical output.
- 4) Columns 7 - 9 - blank
- 5) Columns 10 - 11 (Format I2) - Maximum number of characters in any one line of information (Maximum 12 characters)
- 6) Columns 12 - 23 (Format 3(A4)) - First line of information
- 7) Columns 24 - 25 - blank
- 8) Columns 26 - 37 (Format 3(A4)) - Second line of information

- 9) Columns 38 - 39 - blank
- 10) Columns 40 - 51 (Format 3(A4)) - Third line of information
- 11) Columns 52 - 53 - blank
- 12) Columns 54 - 65 (Format 3(A4)) - Fourth line of information
- 13) Columns 66 - 67 - blank
- 14) Columns 68 - 79 (Format 3(A4)) - Fifth line of information

NOTE: ALL ALPHA-NUMERIC INFORMATION IS LEFT
JUSTIFIED.....

- b) All 8000 codes refer to IMD information
- c) All SAL codes and Unattended Experiment codes must be greater than 8000 and less than 9999.
- d) All crewman activities file cards must be kept in sequential order.

APPENDIX C

COST-EFFECTIVENESS SUPPORT CALCULATIONS

I. Calculation of Systems Expected Value for Alternative I.

$$E(E)_i = \sum W_i E(P_i) C(P_i) \quad E(R)Cr \quad E(V)Cv \quad / \quad \sum W_i$$

1) Calculation of $\sum W_i E(P_i) C(P_i)$

Effectiveness Parameter	C(Pi)*	Wi	E(Pi)	WiE(Pi)C(Pi)
Function	.90	.50	.15	.0675
Accuracy	.90	.30	.75	.2025
Adaptability	.90	.20	.50	.0900

$$\sum W_i = 1.00 \quad \sum W_i E(P_i) C(P_i) = .3600$$

2) Calculation of $E(R)Cr$

$$E(R)Cr = .75(.90) = .675$$

3) Calculation of $E(V)Cv$

$$E(V)Cv = .30(.90) = .27$$

4) Calculation of Alternative I expected value

$$E(E)_i = .36(.675)(.27) / 1.0 = .0656$$

II. Calculation of Systems Expected Value for Alternative II.

1) Calculation of $\sum W_i E(P_i) C(P_i)$

Effectiveness Parameter	C(Pi)*	Wi	E(Pi)	WiE(Pi)C(Pi)
Function	.90	.50	.85	.3825
Accuracy	.90	.30	.95	.2565
Adaptability	.90	.20	.75	.1350

$$\sum W_i = 1.00 \quad \sum W_i E(P_i) C(P_i) = .7740$$

2) Calculation of $E(R)Cr$

$$E(R)Cr = .75(.90) = .675$$

3) Calculation of $E(V)Cv$

$$E(V)Cv = .95(.90) = .855$$

4) Calculation of Alternative II expected value

$$E(E)ii = .744(.675)(.855) / 1.00 = .4460$$

*C(P) Confidence level assumed to be 90% throughout analysis

APPENDIX D - NEWTIME Program

```

PROGRAM NEWTIME (INPUT, OUTPUT, TAPE2=INPUT, TAPE3=OUTPUT, TAPE4, TAPE5,
1 TAPE6, TAPE7, TAPE8, TAPE9, TAPE10)
COMMON/DATMON/DOY, ANEWMN, YEAR, DOWEEK
COMMON/HEADER/EDITN, GEHRST, GEMNST, GEHRSP, GEMNSP, AMD, DOYD, MISSON
1, PUBDAT1, PUBDAT2, GEDAST, GEDASP
COMMON/BETANGL/GETREF1, GETSPR1, ITIMTRU, ITRORB, ATM, ALAUNCH
DIMENSION Ibuff (1024)
CALL PLOTSC (Ibuff, 1024)
INPT=2
K=1
ATM=1.
ITIMTRU=1
ITRORB=1
1000 FORMAT (A10, A10, F5.8, F5.0, F5.1, F5.0, F5.0, F5.1, I5, I5, A10)
2000 FORMAT (2F10.2, 2A10, F10.0, 2I5, A5)
3000 FORMAT (5X, *MISSION DAY*, 2X, F4.0, 1X, *HAS BEEN SUCCESSFULLY PLOTTED*)
READ (INPT, 1000) TIMFRM, EDITN, GEDAST, GEHRST, GEMNST, GEDASP, GEHRSP,
1 GEMNSP, MD, IDOY, MISSON
READ (INPT, 2000) ANEWMN, GETREF1, PUBDAT1, PUBDAT2, YEAR, LOWDA, IHIDA
*, DOWEEK
ALAUNCH=GETREF1+(ABS (GEDAST)*1440.)+(ABS (GEHRST)*60.)+(ABS (GEMNST)
*)
AMD=FLOAT (MD)
DOY=FLOAT (IDOY)
DOYD=DOY
GETSPR1=GETREF1+1440.
11 IF (ITIMTRU.LT.LOWDA) GO TO 30
10 ICHK=K-2*(K/2)
IF (ICHK-1) 1, 2, 1
1 CALL PLOTCC (0.0, -14.00, 23)
GO TO 3
2 CALL PLOTCC (10.5, 29.3, 23)
3 CALL PLOTCC (0., -.25, -3)
CALL FRAM1
IF (TIMFRM.EQ.3HSTD) GO TO 4
CALL DAYSAY
GO TO 5
4 CALL STAND
5 CONTINUE
CALL HEADER
CALL DATMON
CALL BETANGL
CALL REVOLTV
CALL DANIT
CALL MOONSET
CALL MSFNCV
CALL ORBUN
CALL WRLDHAP
30 DOY=DOY+1.
GEHRST=GEHRSP
GEMNST=GEMNSP
GEDAST=GEDASP
GEDASP=GEDAST+1.
WRITE (3, 3000) AMD
DOYD=DOY
AMD=AMD+1.
ITIMTRU=ITIMTRU+1
GETREF1=GETSPR1
GETSPR1=GETREF1+1440.
K=K+1
IF (ITIMTRU.GE.IHIDA) 20, 11

```

```

20 CALL PLOTCC(10.0,0.0,999)
  CALL EXIT
  END
  SUBROUTINE FRAM1

```

```

C
C SUBROUTINE FRAM1 CONSTRUCTS THE TIMELINE FRAMES AND ENTERS THE STANDARD
C HOLLORITH INFORMATION ON FRAME.

```

```

C
C CONSTRUCTION OF OUTSIDE OF FRAME

```

```

C
  CALL PLOTCC(0.,-14.,2)
  CALL PLOTCC(9.94,-14.,2)
  CALL PLOTCC(9.94,0.,2)
  CALL PLOTCC(0.,0.,2)

```

```

C
C CONSTRUCTION OF HORIZONTAL LINES WITHIN FRAME

```

```

C
  CALL PLOTCC(2.5,0.,3)
  CALL PLOTCC(2.5,-13.0,2)
  CALL PLOTCC(3.75,-13.0,3)
  CALL PLOTCC(3.75,0.,2)
  CALL PLOTCC(4.00,-.375,3)
  CALL PLOTCC(4.00,-13.0,2)
  CALL PLOTCC(4.25,-13.0,3)
  CALL PLOTCC(4.25,0.,2)
  CALL PLOTCC(4.50,0.,3)
  CALL PLOTCC(4.50,-13.0,2)
  CALL PLOTCC(5.374,-13.0,3)
  CALL PLOTCC(5.374,0.,2)
  CALL PLOTCC(5.657,-1.0,3)
  CALL PLOTCC(5.657,-13.0,2)
  CALL PLOTCC(5.94,-13.0,3)
  CALL PLOTCC(5.94,0.,2)
  CALL PLOTCC(6.19,0.,3)
  CALL PLOTCC(6.19,-13.0,2)
  CALL PLOTCC(6.44,-13.0,3)
  CALL PLOTCC(6.44,0.0,2)
  CALL PLOTCC(7.19,-13.0,3)
  CALL PLOTCC(7.19,-.5,2)
  CALL PLOTCC(7.94,-.5,3)
  CALL PLOTCC(7.94,-13.0,2)
  CALL PLOTCC(8.69,-14.,3)
  CALL PLOTCC(8.69,0.,2)
  CALL PLOTCC(9.19,0.,3)
  CALL PLOTCC(9.19,-14.,2)
  CALL PLOTCC(9.44,-14.,3)
  CALL PLOTCC(9.44,0.,2)
  CALL PLOTCC(9.69,0.,3)
  CALL PLOTCC(9.69,-6.562,2)
  CALL PLOTCC(9.69,-8.875,3)
  CALL PLOTCC(9.69,-14.,2)

```

```

C
C CONSTRUCTION OF VERTICAL LINES WITHIN FRAME

```

```

C
  CALL PLOTCC(0.0,-1.0,3)
  CALL PLOTCC(9.19,-1.0,2)
  CALL PLOTCC(8.69,-.5,3)
  CALL PLOTCC(6.44,-.5,2)
  CALL PLOTCC(4.25,-.375,3)
  CALL PLOTCC(3.75,-.375,2)
  CALL PLOTCC(9.94,-1.813,3)
  CALL PLOTCC(9.44,-1.813,2)

```



```

CALL PLOTCC(9.94,-3.125,3)
CALL PLOTCC(9.44,-3.125,2)
CALL PLOTCC(9.94,-5.25,3)
CALL PLOTCC(9.44,-5.25,2)
CALL PLOTCC(9.94,-6.562,3)
CALL PLOTCC(9.44,-6.562,2)
CALL PLOTCC(9.94,-7.625,3)
CALL PLOTCC(9.44,-7.625,2)
CALL PLOTCC(9.94,-8.125,3)
CALL PLOTCC(9.44,-8.125,2)
CALL PLOTCC(9.94,-8.875,3)
CALL PLOTCC(9.44,-8.875,2)
CALL PLOTCC(9.94,-10.563,3)
CALL PLOTCC(9.44,-10.563,2)
CALL PLOTCC(9.94,-11.875,3)
CALL PLOTCC(9.44,-11.875,2)
CALL PLOTCC(0.,-13.0,3)
CALL PLOTCC(8.69,-13.0,2)
YPAGE=-12.5
DO 10 I=1,23
CALL PLOTCC(8.74,YPAGE,3)
CALL PLOTCC(8.64,YPAGE,2)
CALL PLOTCC(6.49,YPAGE,3)
CALL PLOTCC(6.39,YPAGE,2)
CALL PLOTCC(5.374,YPAGE,3)
CALL PLOTCC(5.324,YPAGE,2)
CALL PLOTCC(4.30,YPAGE,3)
CALL PLOTCC(4.20,YPAGE,2)
CALL PLOTCC(3.80,YPAGE,3)
CALL PLOTCC(3.70,YPAGE,2)
CALL PLOTCC(2.550,YPAGE,3)
CALL PLOTCC(2.450,YPAGE,2)
10 YPAGE=YPAGE+.5

```

```

C
C HOLLORITH INFORMATION ONTO FRAME
C

```

```

CALL SYMBAL(9.765,-.81,.1,3HGET,270.,3)
CALL SYMBAL(9.765,-2.200,.1,6HMO/DOY,270.,6)
CALL SYMBAL(9.765,-3.585,.1,12HHOUSTON DATE,270.,12)
CALL SYMBAL(9.765,-5.500,.1,4HBETA,270.,4)
CALL SYMBAL(9.765,-5.95,.1,5HANGLE,270.,5)
CALL SYMBAL(9.765,-6.860,.1,4HMOON,270.,4)
CALL SYMBAL(9.515,-6.860,.1,5HPHASE,270.,5)
CALL SYMBAL(9.765,-9.319,.1,7HMISSION,270.,7)
CALL SYMBAL(9.765,-10.819,.1,7HEDITION,270.,7)
CALL SYMBAL(9.765,-12.135,.1,16HPUBLICATION DATE,270.,16)
CALL SYMBAL(9.265,-.1,.1,3HREV,270.,3)
CALL SYMBAL(8.790,-.1,.2,3HGMT,270.,3)
CALL SYMBAL(7.49,-.1,.1,3HCMN,270.,3)
CALL SYMBAL(8.25,-.6,.1,3HCDR,270.,3)
CALL SYMBAL(7.49,-.6,.1,3HSPT,270.,3)
CALL SYMBAL(6.74,-.6,.1,3HPLT,270.,3)
CALL SYMBAL(6.265,-.1,.1,9HDAY/NIGHT,270.,9)
CALL SYMBAL(6.015,-.1,.1,10HMOON UP/DN,270.,10)
CALL SYMBAL(5.674,-.05,.1,11HEARTH TRACE,270.,11)
CALL SYMBAL(5.474,-.3,.1,5HW/SAA,270.,5)
CALL SYMBAL(5.0,-.1,.1,7HNETWORK,270.,7)
CALL SYMBAL(4.8,-.1,.1,8HCOVERAGE,270.,8)
CALL SYMBAL(4.325,-.4,.1,3HIMD,270.,3)
CALL SYMBAL(3.925,-.1,.1,3HSAL,270.,3)
CALL SYMBAL(4.075,-.6,.1,3HSOL,270.,3)
CALL SYMBAL(3.825,-.6,.1,5HA-SOL,270.,5)

```

```

CALL SYMBAL(3.3,-.05,.1,10HUNATTENDED,270.,10)
CALL SYMBAL(2.9,-.05,.1,11HEXPERIMENTS,270.,11)
CALL SYMBAL(1.2,-.3,.1,5HNOTES,270.,5)
RETURN
END
SUBROUTINE DAYSAV
C
C SUBROUTINE DAYSAV ENTERS CORRECT GMT HOURS ON FRAME FOR DAYLIGHT SAVINGS
C TIME MISSIONS
C
COMMON/HEADER/EDITN,GEHRST,GEHNST,GEHRSP,GEMNSP,AMD,DOYD,MISSE
1,PUBDAT1,PUBDAT2,GEDAST,GEDASP
C
C DAY OF THE YEAR BREAKPOINT
C
CALL PLOTCC(9.19,-7.50,3)
CALL PLOTCC(8.99,-7.50,2)
DOYDE=DOYD+1.
CALL NUMBAR(9.04,-7.20,.1,DOYD,270.,-1)
CALL NUMBAR(9.04,-7.55,.1,DOYDE,270.,-1)
C
C MISSION DAY BREAKPOINT
C
CALL PLOTCC(9.19,-10.0,3)
CALL PLOTCC(8.99,-10.0,2)
AMDE=AMD+1.
CALL NUMBAR(9.04,-9.8,.1,AMDE,270.,-1)
CALL NUMBAR(9.04,-10.05,.1,AMDE,270.,-1)
HOUR=12.
YS=1.35
DO 10 I=1,13
CALL NUMBAR(8.79,-YS,.2,HOUR,270.,-1)
HOUR=HOUR+1.
10 YS=YS+.5
HOUR=1.
YS=7.94
DO 20 I=1,9
CALL NUMBAR(8.79,-YS,.2,HOUR,270.,-1)
HOUR=HOUR+1.
20 YS=YS+.5
CALL NUMBAR(8.79,-12.35,.2,10.,270.,-1)
CALL NUMBAR(8.79,-12.85,.2,11.,270.,-1)
RETURN
END
SUBROUTINE STAND
C
C SUBROUTINE STAND ENTERS CORRECT GMT HOURS ON FRAME FOR STANDARD TIME
C MISSIONS
C
COMMON/HEADER/EDITN,GEHRST,GEHNST,GEHRSP,GEMNSP,AMD,DOYD,MISSE
1,PUBDAT1,PUBDAT2,GEDAST,GEDASP
C
C DAY OF THE YEAR BREAKPOINT
C
CALL PLOTCC(9.19,-7.00,3)
CALL PLOTCC(8.99,-7.00,2)
DOYDE=DOYD+1.
CALL NUMBAR(9.04,-6.70,.1,DOYD,270.,-1)
CALL NUMBAR(9.04,-7.05,.1,DOYDE,270.,-1)
C
C MISSION DAY BREAKPOINT
C

```

```

CALL PLOTCC(9.19,-10.0,3)
CALL PLOTCC(8.99,-10.0,2)
AMDE=AMD+1.
CALL NUMBAR(9.04,-9.8,.1,AMD,270.,-1)
CALL NUMBAR(9.04,-10.05,.1,AMDE,270.,-1)
HOUR=13.
YS=1.35
DO 10 I=1,12
CALL NUMBAR(8.79,-YS,.2,HOUR,270.,-1)
HOUR=HOUR+1.
10 YS=YS+.5
HOUR=1.
YS=7.44
DO 20 I=1,9
CALL NUMBAR(8.79,-YS,.2,HOUR,270.,-1)
HOUR=HOUR+1.
20 YS=YS+.5
CALL NUMBAR(8.79,-11.85,.2,10.,270.,-1)
CALL NUMBAR(8.79,-12.35,.2,11.,270.,-1)
CALL NUMBAR(8.79,-12.85,.2,12.,270.,-1)
RETURN
END
SUBROUTINE HEADER
C
C THIS SUBROUTINE PLACES GET,MD/DOY,MISSION AND EDITION INFORMATION ON
C TIMELINE FRAME
C
COMMON/HEADER/EDITN,GEHRST,GEMNST,GEHRSP,GEMNSP,AMD,DOYD,MISSE
1,PUBDAT1,PUBDAT2,GEDAST,GEDASP
C
C GET DAYS-HOURS-MINUTES
C
CALL NUMBAR(9.49,-.1125,.1,GEDAST,270.,-1)
CALL SYMBAL(9.49,-.288,.15,78,270.,-1)
CALL NUMBAR(9.49,-.3855,.1,GEHRST,270.,-1)
CALL SYMBAL(9.49,-.5610,.15,78,270.,-1)
CALL NUMBAR(9.49,-.6585,.1,GEMNST,270.,-1)
CALL SYMBAL(9.49,-.834,.1,67,270.,-1)
CALL NUMBAR(9.49,-.9215,.1,GEDASP,270.,-1)
CALL SYMBAL(9.49,-1.097,.15,78,270.,-1)
CALL NUMBAR(9.49,-1.1945,.1,GEHRSP,270.,-1)
CALL SYMBAL(9.49,-1.37,.15,78,270.,-1)
CALL NUMBAR(9.49,-1.4675,.1,GEMNSP,270.,-1)
C
C MD/DOY
C
CALL NUMBAR(9.49,-2.2000,.1,AMD,270.,-1)
CALL SYMBAL(9.49,-2.3750,.1,67,270.,-1)
CALL NUMBAR(9.49,-2.4625,.1,DOYD,270.,-1)
C
C MISSION
C
CALL SYMBAL(9.49,-9.3190,.1,MISSE,270.,10)
C
C EDITION
C
CALL SYMBAL(9.49,-10.8190,.1,EDITN,270.,10)
C
C PUBLICATION DATE
C
CALL SYMBAL(9.49,-12.135,.1,PUBDAT1,270.,10)
CALL SYMBAL(9.49,-12.995,.1,PUBDAT2,270.,10)

```

```

      RETURN
      END
      SUBROUTINE DATMON
C
C SUBROUTINE DATMON ENTERS THE HOUSTON DATE AND MOON PHASE ON TIMELINE
C
      COMMON/DATMON/DOY,ANEWMN,YEAR,DOWEEK
C
C HOUSTON DATE DETERMINATION
C
      IDOY=IFIX(DOY)
      CALL SYMBAL(9.49,-3.2375,.1,DOWEEK,270.,4)
      IF(DOWEEK.EQ.4HMON.) GO TO 200
      IF(DOWEEK.EQ.4HTUE.) GO TO 201
      IF(DOWEEK.EQ.4HWED.) GO TO 202
      IF(DOWEEK.EQ.4HTHU.) GO TO 203
      IF(DOWEEK.EQ.4HFRI.) GO TO 204
      IF(DOWEEK.EQ.4HSAT.) GO TO 205
      IF(DOWEEK.EQ.4HSUN.) GO TO 206
200 DOWEEK=4HTUE.
      GO TO 210
201 DOWEEK=4HWED.
      GO TO 210
202 DOWEEK=4HTHJ.
      GO TO 210
203 DOWEEK=4HFRI.
      GO TO 210
204 DOWEEK=4HSAT.
      GO TO 210
205 DOWEEK=4HSUN.
      GO TO 210
206 DOWEEK=4HMON.
210 CONTINUE
      IF(IDOY-31)7,7,8
      7 CALL SYMBAL(9.49,-3.6375,.1,7HJANUARY,270.,7)
      CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
      GO TO 60
      8 IF(IDOY-59)9,9,10
      9 CALL SYMBAL(9.49,-3.6375,.1,8HFEBRUARY,270.,8)
      DOY=DOY-31.
      CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
      DOY=DOY+31.
      GO TO 60
10 IF(IDOY-90)31,31,32
31 CALL SYMBAL(9.49,-3.6375,.1,5HMARCH,270.,5)
      DOY=DOY-59.
      CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
      DOY=DOY+59.
      GO TO 60
32 IF(IDOY-120)33,33,34
33 CALL SYMBAL(9.49,-3.6375,.1,5HAPRIL,270.,5)
      DOY=DOY-90.
      CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
      DOY=DOY+90.
      GO TO 60
34 IF(IDOY-151)35,35,36
35 CALL SYMBAL(9.49,-3.6375,.1,3HMAY ,270.,3)
      DOY=DOY-120.
      CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
      DOY=DOY+120.
      GO TO 60
36 IF(IDOY-181)37,37,38

```

```

37 CALL SYMBAL(9.49,-3.6375,.1,4HJUNE ,270.,4)
   DOY=DOY-151.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+151.
   GO TO 60
38 IF(IDOY-212) 39,39,40
39 CALL SYMBAL(9.49,-3.6375,.1,4HJULY ,270.,4)
   DOY=DOY-181.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+181.
   GO TO 60
40 IF(IDOY-243) 41,41,42
41 CALL SYMBAL(9.49,-3.6375,.1,6HAUGUST,270.,6)
   DOY=DOY-212.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+212.
   GO TO 60
42 IF(IDOY-273) 43,43,44
43 CALL SYMBAL(9.49,-3.6375,.1,9HSEPTEMBER,270.,9)
   DOY=DOY-243.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+243.
   GO TO 60
44 IF(IDOY-304) 45,45,46
45 CALL SYMBAL(9.49,-3.6375,.1,7HOCTOBER ,270.,7)
   DOY=DOY-273.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+273.
   GO TO 60
46 IF(IDOY-334) 47,47,48
47 CALL SYMBAL(9.49,-3.6375,.1,8HNOVEMBER ,270.,8)
   DOY=DOY-304.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+304.
   GO TO 60
48 CONTINUE
   CALL SYMBAL(9.49,-3.6375,.1,8HDECEMBER ,270.,8)
   DOY=DOY-334.
   CALL NUMBAR(9.49,-4.5125,.1,DOY,270.,-1)
   DOY=DOY+334.
60 CALL SYMBAL(9.49,-4.6875,.1,1H,,270.,5)
   CALL NUMBAR(9.49,-4.77,.1,YEAR,270.,-1)

```

C

C MOON PORTION

C

```

   CALL PLOTCC(9.69,-7.875,23)
   CALL FACTOR(.2)
   FRDAY=DOY+.5
   TDAY=FRDAY-ANEWMN
   IF(TDAY-29.530589) 50,50,51
51 IF(TDAY-59.061178) 52,52,53
52 TDAY=TDAY-29.530589
   GO TO 50
53 TDAY=TDAY-59.061178
50 DO 111 I=10,360,10
   RAD=FLOAT(I)*(3.14159/180.)
   XCIR=COS(RAD)
   YCIR=SIN(RAD)
   IF(I-10) 112,112,113
112 CALL PLOTCC(1.,0.,3)
   CALL PLOTCC(XCIR,YCIR,2)
   GO TO 111

```

```

113 CALL PLOTCC(XCIR,YCIR,2)
111 CONTINUE
    THETA=(TDAY/29.530589)*360.
    RADTH=THETA*(3.14159/180.)
    CALL PLOTCC(1.,0.,3)
    IF(THETA-180.)14,14,15
  14 PHI=60.
118 RADPH=PHI*(3.14159/180.)
    XDST=SIN(RADPH)
    YDST=-COS(RADPH)*COS(RADTH)
    CALL PLOTCC(XDST,YDST,2)
    YTERM=COS(RADPH)
    CALL PLOTCC(XDST,YTERM,2)
    CALL PLOTCC(XDST,YDST,2)
    IF(PHI-0.)116,116,117
117 PHI=PHI-30.
    GO TO 118
116 PHI=150.
  11 RADPH=PHI*(3.14159/180.)
    XDST=-SIN(RADPH)
    YDST=COS(RADPH)*COS(RADTH)
    CALL PLOTCC(XDST,YDST,2)
    YTERM=-COS(RADPH)
    CALL PLOTCC(XDST,YTERM,2)
    CALL PLOTCC(XDST,YDST,2)
    IF(PHI-120.)110,110,119
119 PHI=PHI-30.
    GO TO 11
  15 PHI=60.
  16 RADPH=PHI*(3.14159/180.)
    XDST=SIN(RADPH)
    YDST=COS(RADPH)*COS(RADTH)
    CALL PLOTCC(XDST,YDST,2)
    YTERM=-COS(RADPH)
    CALL PLOTCC(XDST,YTERM,2)
    CALL PLOTCC(XDST,YDST,2)
    IF(PHI-0.)17,17,18
  18 PHI=PHI-30.
    GO TO 16
  17 PHI=150.
  29 RADPH=PHI*(3.14159/180.)
    XDST=-SIN(RADPH)
    YDST=-COS(RADPH)*COS(RADTH)
    CALL PLOTCC(XDST,YDST,2)
    YTERM=COS(RADPH)
    CALL PLOTCC(XDST,YTERM,2)
    CALL PLOTCC(XDST,YDST,2)
    IF(PHI-120.)110,110,120
120 PHI=PHI-30.
    GO TO 19
110 CALL PLOTCC(-1.,0.,2)
    CALL FACTOR(1.0)
    CALL PLOTCC(-9.69,7.875,23)
    RETURN
  END
  SUBROUTINE BETANGL
C
C THIS SUBROUTINE INTERPOLATES BETWEEN TWO BETA ANGLES 24 HOURS APART
C AS GIVEN ON THE MPAD TAPE AND ENTERS THE INFORMATION ON THE TIMELINE
C FRAME
C
COMMON/BETANGL/GETREF1,GETSPR1

```

```

1000 FORMAT(I3,1X,I2,1X,I2,1X,F4.1,F6.2)
3000 FORMAT(50X,A1)
      IBETCO=4
      12 READ(IBETCO,3000)DUMY
      10 READ(IBETCO,1000)IDAYGET,IHORGET,IGETHIN,SECDGET,ANGLE
      IF(EOF,IBETCO)999,11
      11 DAYGET=FLOAT(IDAYGET)
      HOURGET=FLOAT(IHORGET)
      GETMINT=FLOAT(IGETHIN)
      BETAMIN=(DAYGET*1440.)+(HOURGET*60.)+GETMINT+(SECDGET/60.)
      IF(GETREF1-BETAMIN)30,30,10
      30 IF(GETSPR1-BETAMIN)42,20,20
      20 DIST=(BETAMIN-GETREF1)*.008333333333+1.
      CALL NUMBAR(9.49,-5.60,.1,ANGLE,270.,1)
      CALL SYMBAL(9.04,-DIST,.15,18,270.,-1)
      DIST=DIST+.1
      CALL SYMBAL(9.04,-DIST,.1,29,270.,-1)
      42 REWIND IBETCO
      RETURN
999 CALL PLOTCC(10.0,0.0,999)
      CALL EXIT
      END
      SUBROUTINE REVOLTN

```

C
C SUBROUTINE REVOLTN READS REVOLUTION TIMES FROM MPAD TAPE, PLOTS THEM ON
C TIMELINE FRAME IN PROPER LOCATION AND SAVES TIMES BETWEEN REVS FOR
C WORLD MAP PLOTTING
C

```

      COMMON/WRLDMAP/AINIAL(20)
      COMMON/BETANGL/GETREF1,GETSPR1
1000 FORMAT(F6.1,I3,1X,I2,1X,I2,1X,F4.1)
2000 FORMAT(50X,A1)
      IREVCO=5
      REWIND IREVCO
      READ(IREVCO,2000)DUMY
      I=2
      12 READ(IREVCO,1000)REV,IDAYGET,IHORGET,IGETHIN,SECDGET
      IF(EOF,IREVCO)999,11
      11 DAYGET=FLOAT(IDAYGET)
      HOURGET=FLOAT(IHORGET)
      GETMIN=FLOAT(IGETHIN)
      REVMINT=(DAYGET*1440.)+(HOURGET*60.)+GETMIN+(SECDGET/60.)
      IF(GETREF1-REVMINT)20,20,10
      10 AINIAL(1)=REVMINT
      GO TO 12
      20 AINIAL(I)=REVMINT
      IF(GETSPR1-REVMINT)90,21,21
      21 DIST=((REVMINT-GETREF1)/1440.)*12.
      CALL PLOTCC(9.19,-DIST-1.,3)
      CALL PLOTCC(9.44,-DIST-1.,2)
      CALL PLOTCC(5.94,-DIST-1.,3)
      CALL PLOTCC(5.374,-DIST-1.,2)
      REV=REV+.1
      DIST=DIST+.1
      CALL NUMBAR(9.24,-DIST-1.,.15,REV,270.,-1)
      I=I+1
      GO TO 12
      90 CONTINUE
      AINIAL(I)=-1
      RETURN
999 CALL PLOTCC(10.0,0.0,999)
      CALL EXIT

```

END
SUBROUTINE DANIT

C
C THIS SUBROUTINE READS DAY NIGHT TIMES FROM MPAD TAPE AND PLOTS THEM ON
C THE TIMELINE FRAME IN THE PROPER PLACE
C

```

COMMON/BET ANGL/GETREF1,GETSPR1,IDUMP,IDUMQ,ATM,ALAUNCH
1000 FORMAT(14X,I3,1X,I2,1X,I2,1X,F4.1,I2,1X,F4.1)
      IDANCO=6
      REWIND IDANCO
2000 FORMAT(50X,A1)
      READ(IDANCO,2000)DUMY
13 READ(IDANCO,1000)IDAYSET,IHORSET,IMINSET,SECSET,IDELMIN,DELTSEC
      IF(EOF,IDANCO)999,11
11 DAYSET=FLOAT(IDAYSET)
      HOURSET=FLOAT(IHORSET)
      SETMIN=FLOAT(IMINSET)
      DELTMIN=FLOAT(IDELMIN)
      SETSTAT=(DAYSET*1440.)+(HOURSET*60.)+SETMIN+(SECSET/60.)
      SETSTOP=SETSTAT+DELTMIN+(DELTSEC/60.)
      IF(SETSTAT-GETREF1)10,12,12
10 IF(SETSTOP-GETREF1)13,13,14
14 ANEWDEL=SETSTOP-GETREF1
      DISANDL=(ANEWDEL*.00833)
      DISASTR=1.
      DISASTP=1.+DISANDL
      CALL PLOTCC(6.19,-DISASTR,3)
      GO TO 15
12 ANEWDEL=0.
15 IF(ANEWDEL-0.)16,16,17
20 IF(DISASTR-DISASTP)17,17,13
17 CALL PLOTCC(6.19,-DISASTR,2)
      CALL PLOTCC(6.44,-DISASTR,2)
      DISASTR=DISASTR+.02
      GO TO 20
16 IF(SETSTAT-GETSPR1)50,50,25
50 IF(SETSTOP-GETSPR1)21,21,22
21 DISTST=((SETSTAT-GETREF1)*.00833)+1.
      DISTSP=((SETSTOP-GETREF1)*.00833)+1.
      IF(SETSTAT.LT.ALAUNCH)GO TO 202
      DCHEK=DISTST-.3
      IF(DCHEK.LE.1.0)GO TO 200
      CALL NUMBAR(6.265,-DISTST+.3,.1,ATM,270.,-1)
200 ATM=ATM+1.
202 CALL PLOTCC(6.19,-DISTST,3)
30 IF(DISTST-DISTSP)31,31,13
31 CALL PLOTCC(6.19,-DISTST,2)
      CALL PLOTCC(6.44,-DISTST,2)
      DISTST=DISTST+.02
      GO TO 30
22 CONTINUE
      DISTST=((SETSTAT-GETREF1)*.00833)+1.
      DISTSP=13.
      IF(SETSTAT.LT.ALAUNCH)GO TO 203
      DCHEK=DISTST-.3
      IF(DCHEK.LE.1.0)GO TO 201
      CALL NUMBAR(6.265,-DISTST+.3,.1,ATM,270.,-1)
201 ATM=ATM+1.
203 CALL PLOTCC(6.19,-DISTST,3)
23 IF(DISTST-DISTSP)40,40,25
40 CALL PLOTCC(6.19,-DISTST,2)
      CALL PLOTCC(6.44,-DISTST,2)

```



```

DISTST=DISTST+.02
GO TO 23
25 CONTINUE
RETURN
999 CALL PLOTCC(10.0,0.0,999)
CALL EXIT
END
SUBROUTINE MSFNCV
C
C THIS SUBROUTINE READS THE MSFN STATION START AND STOP TIMES FROM THE
C MPAD TAPE AND ENTERS THEM ON TIMELINE
C
COMMON/BETANGL/GETREF1,GETSPR1
1000 FORMAT(A3,28X,I3,1X,I2,1X,I2,1X,F4.1,14X,I2,1X,F4.1)
IMSNCO=7
2000 FORMAT(50X,A1)
REWIND IMSNCO
READ(IMSNCO,2000)DUMY
L=1
10 READ(IMSNCO,1000)STATON,IDAYACQ,IHORACQ,IMINACQ,SECDACQ,
*IDELMIN,SECDLT
IF(STATON.EQ.3HNF0)GO TO 10
IF(E0F,IMSNCO)999,11
11 DAYACQ=FLOAT(IDAYACQ)
HOURACQ=FLOAT(IHORACQ)
ACQMIN=FLOAT(IMINACQ)
DELMIN=FLOAT(IDELMIN)
STASTAT=(DAYACQ*1440.)+(HOURACQ*60.)+ACQMIN+(SECDACQ/60.)
STASTOP=STASTAT+DELMIN+(SECDLT/60.)
DINCER=DELMIN+(SECDLT/60.)
IF(DINCER.LE.2.0)GO TO 10
IF(STASTAT-GETREF1)10,20,20
20 IF(STASTAT-GETSPR1)21,21,90
21 DISTST=((STASTAT-GETREF1)*.00833)+1.
DISTSP=((STASTOP-GETREF1)*.00833)+1.
IF(DISTSP-13.)30,30,40
40 DISTSP=13.
30 IF(L-2)890,900,910
890 CALL PLOTCC(5.2365,-DISTST,3)
CALL PLOTCC(5.2365,-DISTSP,2)
XDIST=5.199
920 DISTSP=DISTSP+.0375
CALL PLOTCC(XDIST,-DISTSP,3)
CALL SYMBAL(XDIST,-DISTSP,.075,STATON,270.,3)
GO TO 980
980 CALL PLOTCC(5.1115,-DISTST,3)
CALL PLOTCC(5.1115,-DISTSP,2)
XDIST=5.074
GO TO 920
910 IF(L-4)1100,1110,1120
1100 CALL PLOTCC(4.9865,-DISTST,3)
CALL PLOTCC(4.9865,-DISTSP,2)
XDIST=4.949
GO TO 920
1110 CALL PLOTCC(4.8615,-DISTST,3)
CALL PLOTCC(4.8615,-DISTSP,2)
XDIST=4.824
GO TO 920
1120 IF(L-5)1130,1130,1140
1130 CALL PLOTCC(4.7365,-DISTST,3)
CALL PLOTCC(4.7365,-DISTSP,2)
XDIST=4.699

```

```

      GO TO 920
1140 CALL PLOTCC(4.6115,-DISTST,3)
      CALL PLOTCC(4.6115,-DISTSP,2)
      XDIST=4.574
      GO TO 920
980  L=L+1
      IF(L-6) 10,10,990
990  L=1
      GO TO 10
90   RETURN
999  CALL PLOTCC(10.0,0.0,999)
      CALL EXIT
      END
      SUBROUTINE ORBUN

```

C
C SUBROUTINE ORBUN READS LATITUDE AND LONGITUDE INFORMATION FROM THE
C MPAD TAPE AND PLOTS THE ORBIT TRACE ON TIMELINE
C

```

      COMMON/BETANGL/GETREF1,GETSPR1,ITIMTRU,ITRORB
1000 FORMAT(I3,1X,I2,1X,I2,1X,F4.1,10X,F6.2)
2000 FORMAT(50X,A1)
      IORBCO=8
      L=1
      IF(ITRORB-1) 10,13,10
13   REWIND IORBCO
      READ(IORBCO,2000)DUMY
10   READ(IORBCO,1000)IDAYLAT,IHORLAT,IMINLAT,SECOLAT,ALAT
      IF(EOF,IORBCO)999,11
11   DAYLAT=FLOAT(IDAYLAT)
      HOURLAT=FLOAT(IHORLAT)
      AMINLAT=FLOAT(IMINLAT)
      POINTLT=(DAYLAT*1440.)+(HOURLAT*60.)+AMINLAT+(SECOLAT/60.)
      IF(POINTLT-GETREF1) 10,20,20
20   IF(POINTLT-GETSPR1) 21,21,90
21   DISTY=((POINTLT-GETREF1)*.00833)+1.
      DEG2INC=ALAT*.00314444444
      DISTX=5.657+DEG2INC
      IF(L-1) 23,22,23
22   CALL PLOTCC(DISTX,-DISTY,3)
      L=L+1
      GO TO 10
23   CALL PLOTCC(DISTX,-DISTY,2)
      GO TO 10
90   ITRORB=ITRORB+1
      RETURN
999  CALL PLOTCC(10.0,0.0,999)
      CALL EXIT
      END
      SUBROUTINE WRDMAP

```

C
C SUBROUTINE WRDMAP PLOTS THE OUTLINE OF THE CONTINENTS ON THE
C TIMELINE FRAME IN THE PROPER POSITION TO COINCIDE WITH THE REV
C COUNTER
C

```

1000 FORMAT(I3)
2000 FORMAT(2F10.4)
      COMMON/WRDMAP/AINIAL(20)
      COMMON/BETANGL/GETREF1,GETSPR1
      IWROCO=9
      IEND=1
      IF(AINIAL(2)-0.0)90,90,910
910  IF(AINIAL(1)-0.0)11,11,12

```

```

11 AINIAL(1)=-AINIAL(3)+AINIAL(2)+AINIAL(2)
12 FST800G=AINIAL(1)-GETREF1
   IF(FST800G-0.0)13,13,14
13 IJK=2
   FST800G=AINIAL(2)-GETREF1
   GO TO 15
14 IJK=1
15 REWIND IWROCO
   FS800ST=(FST800G*.008333)+1.
   IF(AINIAL(IJK+1)-0.0)31,31,30
31 AINIAL(IJK+1)=AINIAL(IJK)+AINIAL(IJK)-AINIAL(IJK-1)
   IEND=999
30 VARDIST=((AINIAL(IJK+1)-AINIAL(IJK))*0.00833)/360.
   DO 10 I=1,10
   READ(IWROCO,1000)NOP
   IPOINT=1
   DO 20 JI=1,NOP
   READ(IWROCO,2000)ALAT,ALONG
   DEG2INC=ALAT*.0031444444
   ALATDST=5.657+DEG2INC
   ALONDST=((ALONG-280.)*VARDIST)+FS800ST
   IF(IPOINT-1)40,41,40
41 CALL PLOTCC(ALATDST,-ALONDST,3)
   IPOINT=IPOINT+1
40 IF(ALONDST-1.)22,22,23
23 IF(ALONDST-13.)25,25,22
22 CALL PLOTCC(ALATDST,-ALONDST,3)
   GO TO 24
25 CALL PLOTCC(ALATDST,-ALONDST,2)
24 CONTINUE
20 CONTINUE
10 CONTINUE
   IJK=IJK+1
   FST800G=AINIAL(IJK)-GETREF1
   IF(IEND-999)15,95,95
95 REWIND IWROCO
   DO 96 II=1,10
   READ(IWROCO,1000)NOP
   IPOINT=1
   DO 97 IJI=1,NOP
   READ(IWROCO,2000)ALAT,ALONG
   DEG2INC=ALAT*.00314444444444
   ALATDST=5.657+DEG2INC
   FS800ST=(FST800G*.008333)+1.
   ALONDST=((ALONG-280.)*VARDIST)+FS800ST
   IF(IPOINT-1)140,141,140
141 CALL PLOTCC(ALATDST,-ALONDST,3)
   IPOINT=IPOINT+1
140 IF(ALONDST-13.)98,98,99
99 CALL PLOTCC(ALATDST,-ALONDST,3)
   GO TO 100
98 CALL PLOTCC(ALATDST,-ALONDST,2)
100 CONTINUE
97 CONTINUE
96 CONTINUE
90 RETURN
   END
   SUBROUTINE MOONSET
C
C THIS SUBROUTINE READS MOON RISE SET TIMES FROM MPAD TAPE AND PLOTS
C THEM ON THE TIMELINE FRAME IN THE PROPER PLACE
C

```

```

COMMON/BETANGL/GETREF1,GETSPR1,IDUMP,IDUMQ,ATM
1000 FORMAT(14X,I3,1X,I2,1X,I2,1X,F4.1,I2,1X,F4.1)
      IDANCO=10
      REWIND IDANCO
2000 FORMAT(50X,A1)
      READ(IDANCO,2000)DUMY
13 READ(IDANCO,1000)IDAYSET,IHORSET,IMINSET,SECSET,IDELMIN,DELTSEC
      IF(EOF,IDANCO)999,11
11 DAYSET=FLOAT(IDAYSET)
      HOURSET=FLOAT(IHORSET)
      SETMIN=FLOAT(IMINSET)
      DELTMIN=FLOAT(IDELMIN)
      SETSTAT=(DAYSET*1440.)+(HOURSET*60.)+SETMIN+(SECSET/60.)
      SETSTOP=SETSTAT+DELTMIN+(DELTSEC/60.)
      IF(SETSTAT-GETREF1)10,12,12
10 IF(SETSTOP-GETREF1)13,13,14
14 ANEWDEL=SETSTOP-GETREF1
      DISANDL=(ANEWDEL*.00833)
      DISASTR=1.
      DISASTP=1.+DISANDL
      CALL PLOTCC(5.94,-DISASTR,3)
      GO TO 15
12 ANEWDEL=0.
15 IF(ANEWDEL-0.)16,16,17
20 IF(DISASTR-DISASTP)17,17,13
17 CALL PLOTCC(5.94,-DISASTR,2)
      CALL PLOTCC(6.19,-DISASTR,2)
      DISASTR=DISASTR+.045
      GO TO 20
16 IF(SETSTAT-GETSPR1)50,50,25
50 IF(SETSTOP-GETSPR1)21,21,22
21 DISTST=((SETSTAT-GETREF1)*.00833)+1.
      DISTSP=((SETSTOP-GETREF1)*.00833)+1.
      CALL PLOTCC(5.94,-DISTST,3)
30 IF(DISTST-DISTSP)31,31,13
31 CALL PLOTCC(5.94,-DISTST,2)
      CALL PLOTCC(6.19,-DISTST,2)
      DISTST=DISTST+.045
      GO TO 30
22 CONTINUE
      DISTST=((SETSTAT-GETREF1)*.00833)+1.
      DISTSP=13.
      CALL PLOTCC(5.94,-DISTST,3)
23 IF(DISTST-DISTSP)40,40,25
40 CALL PLOTCC(5.94,-DISTST,2)
      CALL PLOTCC(6.19,-DISTST,2)
      DISTST=DISTST+.045
      GO TO 23
25 CONTINUE
      RETURN
999 CALL PLOTCC(10.0,0.0,999)
      CALL EXIT
      END

```

APPENDIX E - NEWCREW Program

```

PROGRAM NEWCREW(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE12,TAPE
*13)
C
C   SET CROSSHAIR 1 INCH ABOVE LOWER LEFT HAND CORNER OF PLT
C   ACTIVITIES BLOCK
C
      DIMENSION Ibuff(1024)
1000 FORMAT(I1,I4,5X,I2,1X,I2,1X,I2,1X,I2,1X,I1,1X,I2)
1700 FORMAT(I3,14 )
1900 FORMAT(1X,40HILLEGAL NUMBER OF DESCRIPTIVE CHARACTERS)
2000 FORMAT(I4,I1,I1,3X,I2,A4,A4,A4,4(2X,A4,A4,A4) )
2100 FORMAT(A4)
2200 FORMAT(15H ERRONEOUS IKEY,           /06H IKEY=,I4,/
*07H START ,F4.0,1X,F3.0,1X,F3.0/
*07H STOP  ,F4.0,1X,F3.0,1X,F3.0/
*04H CM=,I2,7H IKEYB=,I4)
2300 FORMAT(I1)
2400 FORMAT(1X,*UNATTENDED EXPERIMENT CODING ERROR ON MISSION DAY*,1X,
*F3.0,1X,*ERRONEOUS LINE NUMBER*,1X,I3,1X,*START TIME*,1X,I3,1X,I3)
      ICARD=5
      IT20=12
      J=1
      K=1
      L=0
      ICM=3
      N=0
      REWIND IT20
      REWIND 13
C
C   IT DOES NOT MATTER WHAT VALUE DOYST IS
C
      DOYST=001.
      CALL PLOTSC(Ibuff,1024)
      CALL PLOTCC(0.0,0.0,-3)
      READ(ICARD,2300)ISTDS
      IF(ISTDS-1)5000,5001,5001
5000 ELETW=11.
      AIDEN=6.5
      GO TO 30
5001 ELETW=12.
      AIDEN=6.0
      30 ICMA=ICM
      DOYSA=DOYST
      31 M=0
100 READ(ICARD,1000)IAORSO,IKEY,IHRST,ISTMIN,IHRSP,ISPMIN,ICM,IKEYB
      HRST=FLOAT(IHRST)
      STMIN=FLOAT(ISTMIN)
      HRSP=FLOAT(IHRSP)
      SPMIN=FLOAT(ISPMIN)
      IF(Eof,ICARD)900,32
      32 IMODE=IKEY
      IF (ICMA-ICM)710,700,710
700 N=N+1
      GO TO 720
710 N=1
      IF (ICMA-2) 730,740,720
730 CALL PLOTCC (-1.5,0.,23)
      GO TO 720
740 CALL PLOTCC (-.75,0.,23)
720 IF(IKEY-1000) 721,722,722
721 MI=0

```

```

IF(IKEY-100) 761,723,724
761 IKEY=1100
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
723 IKEY=1100
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
724 IF(IKEY-200) 701,725,726
701 IKEY=1100
   HRSP=HRST+1.0
   SPMIN=STMIN
   GO TO 190
725 IKEY=1140
   HRSP=HRST
   SPMIN=STMIN+30.
   GO TO 190
726 IF(IKEY-250) 851,850,850
851 IKEY=1140
   HRSP=HRST
   SPMIN=STMIN+30.0
   GO TO 190
850 IF(IKEY-300) 705,727,728
705 IKEY=1140
   HRSP=HRST
   SPMIN=STMIN+30.
   GO TO 190
727 IKEY=1100
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
728 IF(IKEY-400) 729,729,1730
729 IKEY=1100
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
1730 IF(IKEY-500) 731,731,1731
731 IKEY=1005
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
1731 IF(IKEY-600) 732,732,6732
6732 IF(IKEY-700) 860,733,712
860 IKEY=1156
   HRSP=HRST
   SPMIN=STMIN+30.
   GO TO 190
733 IKEY=1100
   HRSP=HRST+1.
   SPMIN=STMIN
   GO TO 190
712 IF(IKEY-800) 863,713,709
863 IKEY=1090
   HRSP=HRST
   SPMIN=STMIN+15.
   GO TO 190
713 IKEY=1140
   HRSP=HRST
   SPMIN=STMIN+30.
   GO TO 190
709 IF(IKEY-900) 711,711,6733

```

```

711 IKEY=1100
    HRSP=HRST+1.
    SPMIN=STMIN
    GO TO 190
6733 IF(IKEY-1000)746,30,30
746 IKEY=1100
    HRSP=HRST+1.
    SPMIN=STMIN
    GO TO 190
731 IKEY=1100
    HRSP=HRST+1.
    SPMIN=STMIN
    GO TO 190
722 IF(IKEY-8000)180,105,105
C
C      S.A.L. AND IMD DATA
C
105 CONTINUE
    IF(IKEY-8000)120,110,120
110 IF(HRST-ELETW)130,140,140
130 DSTST=AIDEN+(HRST+(STMIN/60.))* .5
    GO TO 150
140 DSTST=(HRST+(STMIN/60.)-ELETW)* .5
150 IF(HRSP-ELETW)160,161,161
160 DSTSP=AIDEN+(HRSP+(SPMIN/60.))* .5
    GO TO 162
161 DSTSP=(HRSP+(SPMIN/60.)-ELETW)* .5
162 IF(DSTSP-12.)163,163,164
164 DSTSP=12.
163 CALL PLOTCC(-1.99,-DSTST,3)
    CALL PLOTCC(-2.14,-DSTST,2)
    CALL PLOTCC(-2.065,-DSTST,2)
    CALL PLOTCC(-2.065,-DSTSP,2)
    CALL PLOTCC(-1.99,-DSTSP,2)
    CALL PLOTCC(-2.14,-DSTSP,2)
    READ(ICARD,1000)IAORSO,IKEY,IHRST,ISTMIN,IHRSP,ISPMIN,ICM
    HRST=FLOAT(IHRST)
    STMIN=FLOAT(ISTMIN)
    HRSP=FLOAT(IHRSP)
    SPMIN=FLOAT(ISPMIN)
    IF(EOF,ICARD)900,195
120 CONTINUE
    IF(IKEY-9999)121,122,122
122 L=0
    DOYST=DOYST+1.
    M=0
    GO TO 180
121 IF(HRST-ELETW)9130,9140,9140
9130 DSTST=AIDEN+(HRST+(STMIN/60.))* .5
    GO TO 9150
9140 DSTST=(HRST+(STMIN/60.)-ELETW)* .5
9150 IF(HRSP-ELETW)9160,9161,9161
9160 DSTSP=AIDEN+(HRSP+(SPMIN/60.))* .5
    GO TO 9162
9161 DSTSP=(HRSP+(SPMIN/60.)-ELETW)* .5
9162 IF(DSTSP-12.)9163,9163,9164
9164 DSTSP=12.
9163 IF(IAORSO-1)123,124,125
125 DISMIO=((DSTSP-DSTST)/2.)+DSTST-.2
    IF(ICM.LT.1.OR.ICM.GT.5)GO TO 99
    GO TO (7000,7001,7003,7004,7005),ICM
99 WRITE(6,2400)DOYST,ICM,IHRST,ISTMIN

```



```

      GO TO 7050
7000 DSTOPX=-2.69
      DSTMDX=-2.815
      DSBTMX=-2.94
      GO TO 7010
7001 DSTOPX=-2.94
      DSTMDX=-3.055
      DSBTMX=-3.19
      GO TO 7010
7003 DSTOPX=-3.19
      DSTMDX=-3.315
      DSBTMX=-3.44
      GO TO 7010
7004 DSTOPX=-3.44
      DSTMDX=-3.555
      DSBTMX=-3.69
      GO TO 7010
7005 DSTOPX=-3.69
      DSTMDX=-3.815
      DSBTMX=-3.94
7010 IF(IKEY.EQ.3001)GO TO 7011
      CALL PLOTCC(DSTOPX,-DSTST,3)
      CALL PLOTCC(DSBTMX,-DSTST,2)
      CALL PLOTCC(DSTMDX,-DSTST,2)
      IF(IAORSO.EQ.4)GO TO 8000
      CALL PLOTCC(DSTMDX,-DISMID,2)
      IF(IAORSO.EQ.3)DISMID=DISMID+1.6
      IF(IAORSO.EQ.3)GO TO 7012
      DISMID=DISMID+.4
7012 CALL PLOTCC(DSTMDX,-DISMID,3)
8000 CALL PLOTCC(DSTMDX,-DSTSP,2)
      CALL PLOTCC(DSTOPX,-DSTSP,2)
      CALL PLOTCC(DSBTMX,-DSTSP,2)
      IF(IAORSO.EQ.3)DSTSP=DISMID-1.5
      IF(IAORSO.EQ.3)GO TO 7910
      IF(IAORSO.EQ.4)DSTSP=DSTSP+.05
      IF(IAORSO.EQ.4)GO TO 7910
      DSTSP=DISMID-.39
7910 READ(IT20,2000) IKEYA,ILN,IANG,NCH,N11,N12,N13,N21,N22,N23,N31,N32,
      *N33,N41,N42,N43,N51,N52,N53
      IF(IKEYA-IKEY)7910,7930,7920
7920 REWIND IT20
      M=M+1
      IF(M-2)7910,290,290
7930 CALL SYMBAL(DSTMDX-.0375,-DSTSP,.075,N11,270.,4)
      DSTSP=DSTSP+.267
      CALL SYMBAL(DSTMDX-.0375,-DSTSP,.075,N12,270.,4)
      DSTSP=DSTSP+.267
      CALL SYMBAL(DSTMDX-.0375,-DSTSP,.075,N13,270.,4)
      GO TO 7050
7011 CALL PLOTCC(DSTOPX,-DSTST,3)
      CALL PLOTCC(DSBTMX,-DSTST,2)
      CALL PLOTCC(DSTMDX,-DSTST,2)
      CALL PLOTCC(DSTMDX,-DSTSP,2)
      CALL PLOTCC(DSTOPX,-DSTSP,2)
      CALL PLOTCC(DSBTMX,-DSTSP,2)
7050 READ(ICARD,1000)IAORSO,IKEY,IHRST,ISTMIN,IHRSP,ISPMIN,ICH
      HRST=FLOAT(IHRST)
      STMIN=FLOAT(ISTMIN)
      HRSP=FLOAT(IHRSP)
      SPMIN=FLOAT(ISPMIN)
      REWIND IT20

```

```

      IF (EOF, ICARD) 900,105
123  DISMID = ((DSTSP-DSTST)/2.)+DSTST-.2
      CALL PLOTCC(-2.24,-DSTST,3)
      CALL PLOTCC(-2.39,-DSTST,2)
      CALL PLOTCC(-2.315,-DSTST,2)
      CALL PLOTCC(-2.315,-DISMID,2)
      DISMID=DISMID+.4
      CALL PLOTCC(-2.315,-DISMID,3)
      CALL PLOTCC(-2.315,-DSTSP,2)
      CALL PLOTCC(-2.24,-DSTSP,2)
      CALL PLOTCC(-2.39,-DSTSP,2)
      DSTSP=DISMID-.39
9910 READ(IT20,2000) IKEYA,ILN,IANG,NCH,N11,N12,N13,N21,N22,N23,N31,N32,
      *N33,N41,N42,N43,N51,N52,N53
      IF(IKEYA-IKEY)9910,9930,9920
9920 REWIND IT20
      M=M+1
      IF (M-2)9910,290,290
9930 CALL SYMBAL(-2.3525,-DSTSP,.075,N11,270.,4)
      DSTSP=DSTSP+.267
      CALL SYMBAL(-2.3525,-DSTSP,.075,N12,270.,4)
      READ(ICARD,1000) IAORSO,IKEY,IHRST,ISTMIN,IHRSP,ISPMIN,ICM
      HRST=FLOAT(IHRST)
      STMIN=FLOAT(ISTMIN)
      HRSP=FLOAT(IHRSP)
      SPMIN=FLOAT(ISPMIN)
      REWIND IT20
      IF (EOF, ICARD) 900,105
124  DISMID = ((DSTSP-DSTST)/2.)+DSTST-.2
      CALL PLOTCC(-2.49,-DSTST,3)
      CALL PLOTCC(-2.64,-DSTST,2)
      CALL PLOTCC(-2.565,-DSTST,2)
      CALL PLOTCC(-2.565,-DISMID,2)
      DISMID=DISMID+.4
      CALL PLOTCC(-2.565,-DISMID,3)
      CALL PLOTCC(-2.565,-DSTSP,2)
      CALL PLOTCC(-2.49,-DSTSP,2)
      CALL PLOTCC(-2.64,-DSTSP,2)
      DSTSP=DISMID-.39
9911 READ(IT20,2000) IKEYA,ILN,IANG,NCH,N11,N12,N13,N21,N22,N23,N31,N32
      *N33,N41,N42,N43,N51,N52,N53
      IF(IKEYA-IKEY)9911,9931,9921
9921 REWIND IT20
      M=M+1
      IF (M-2)9911,290,290
9931 CALL SYMBAL(-2.6025,-DSTSP,.075,N11,270.,4)
      DSTSP=DSTSP+.267
      CALL SYMBAL(-2.6025,-DSTSP,.075,N12,270.,4)
      READ(ICARD,1000) IAORSO,IKEY,IHRST,ISTMIN,IHRSP,ISPMIN,ICM
      HRST=FLOAT(IHRST)
      STMIN=FLOAT(ISTMIN)
      HRSP=FLOAT(IHRSP)
      SPMIN=FLOAT(ISPMIN)
      REWIND IT20
      IF (EOF, ICARD) 900,105

```

C
C
C

NORMAL CM ACTIVITY

C
C
C

180 IF(DOYSA-DOYST)580,190,580

CALCULATION OF START AND STOP BLOCK

```

190 IF (HRST-ELETW) 200,210,210
200 DSTST=AIDEN+ (HRST+(STMIN/60.))* .5
   GO TO 220
210 DSTST=(HRST+ (STMIN/60.)-ELETW)* .5
220 IF (HRSP-ELETW) 230,240,240
230 DSTSP=AIDEN+ (HRSP+(SPMIN/60.))* .5
   GO TO 250
240 DSTSP=(HRSP+ (SPMIN/60.)-ELETW)* .5
250 REWIND      IT 20
   IF (DSTSP-12.) 270,270,260
260 DSTSP=12.
270 READ(IT 20, 20 00) IKEYA, ILN, IANG, NCH, N11, N12, N13, N21, N22, N23, N31,
   *N32, N33, N41, N42, N43, N51, N52, N53
   IF (IKEYA-IKEY) 270,300,280
280 REWIND IT 20
   M=M+1
   IF (M-2) 270,290,290
290 WRITE(      6, 2200) IKEY, DOYST, HRST, STMIN, DOYSP, HRSP, SPMIN,
   *ICM, IKEYB
   M=0
   GO TO 100
300 IF (IKEYA-6720) 320,310,320
C
C      EREP PASS NUMBER
C
310 WRITE(13,1700) IKEYB
   REWIND      13
   READ(13,2100) N31
   REWIND      13
320 IF (N-1) 360,321,360
321 IF (ICM-2) 330,340,350
330 CALL PLOTCC (1.5,0.,23)
   CALL PLOTCC (0.,-DSTST,3)
   CALL PLOTCC (.75,-DSTST,2)
   GO TO 360
340 CALL PLOTCC (.75,0.,23)
   CALL PLOTCC (0.,-DSTST,3)
   CALL PLOTCC (.75,-DSTST,2)
   GO TO 360
350 CALL PLOTCC (0.,-DSTST,3)
   CALL PLOTCC (.75,-DSTST,2)
C
C      CENTERING BASED ON THE NUMBER OF CHARACTERS
C
360 CALL PLOTCC (0.,-DSTST,3)
   CALL PLOTCC (.75,-DSTST,2)
   AMDPT=((DSTSP-DSTST)/2.)+DSTST
   IF (NCH-4) 370,370,380
370 DSTST=AMDPT-FLOAT(NCH)*0.0335
   GO TO 450
380 IF (NCH-7) 390,390,400
390 DSTST=AMDPT- (FLOAT(NCH)*0.0335-0.0005)
   GO TO 450
400 IF (NCH-10) 410,410,420
410 DSTST=AMDPT- (FLOAT(NCH)*0.0335-0.001)
   GO TO 450
420 IF (NCH-12) 430,430,440
430 DSTST=AMDPT- (FLOAT(NCH)*0.0335-0.0015)
   GO TO 450
440 WRITE(      6,1900)
   GO TO 430
450 IF (IANG-1) 460,470,470

```

```

460 ANGL=270.
    GO TO 480
470 ANGL=0.
    IF (HRST-ELETW) 610,620,620
610 DSTST=AIDEN+ (HRST+ (STMIN/60.))*.5
    GO TO 630
620 DSTST= (HRST+ (STMIN/60.))-ELETW)*.5
630 IF (HRSP-ELETW) 640,650,650
640 DSTSP=AIDEN+ (HRSP+ (SPMIN/60.))*.5
    GO TO 660
650 DSTSP= (HRSP+ (SPMIN/60.))-ELETW)*.5
660 REWIND      IT20
    IF (DSTSP-12.) 680,680,670
670 DSTSP=12.
680 CALL PLOTCC (0.,-DSTST,3)
    CALL PLOTCC (2.25,-DSTST,2)
    IF (NCH-4) 3070,3070,3080
3070 XDXST=1.125-FLOAT(NCH)*.0335
    GO TO 4050
3080 IF (NCH-7) 3090,3090,4000
3090 XDXST=1.125- (FLOAT(NCH)*.0335-.0005)
    GO TO 4050
4000 IF (NCH-10) 4010,4010,4020
4010 XDXST=1.125- (FLOAT(NCH)*.0335-.001)
    GO TO 4050
4020 IF (NCH-12) 4030,4030,4040
4030 XDXST=1.125- (FLOAT(NCH)*.0335-.0015)
    GO TO 4050
4040 WRITE(      6,1900)
    GO TO 4030
4050 DSTSP=-DSTSP
    CALL PLOTCC (0.,DSTSP,3)
    CALL PLOTCC (2.25,DSTSP,2)
    DSTSP=DSTSP+.0005
    CALL SYMBAL (XDXST,DSTSP,.075,N11,ANGL,4)
    XDXST=XDXST+.267
    CALL SYMBAL (XDXST,DSTSP,.075,N12,ANGL,4)
    XDXST=XDXST+.267
    CALL SYMBAL (XDXST,DSTSP,.075,N13,ANGL,4)
    GO TO 30
C
C      CENTERING BASED ON THE NUMBER OF LINES
C
480 IF (ILN-2) 490,500,510
490 CALL SYMBAL(.3,-DSTST,.075,N11,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL(.3,-DSTST,.075,N12,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL(.3,-DSTST,.075,N13,ANGL,4)
    GO TO 550
500 CALL SYMBAL(.45,-DSTST,.075,N11,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL(.45,-DSTST,.075,N12,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL(.45,-DSTST,.075,N13,ANGL,4)
    DSTST=DSTST-.534
    CALL SYMBAL (.3,-DSTST,.075,N21,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3,-DSTST,.075,N22,ANGL,4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3,-DSTST,.075,N23,ANGL,4)
    GO TO 550

```

```

510 IF (ILN-4) 520, 530, 540
520 CALL SYMBAL (.425, -DSTST, .075, N11, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N12, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N13, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.3, -DSTST, .075, N21, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N22, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N23, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.175, -DSTST, .075, N31, 270., 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.175, -DSTST, .075, N32, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.175, -DSTST, .075, N33, ANGL, 4)
    GO TO 550
530 CALL SYMBAL (.55, -DSTST, .075, N11, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.55, -DSTST, .075, N12, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.55, -DSTST, .075, N13, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.425, -DSTST, .075, N21, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N22, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N23, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.3, -DSTST, .075, N31, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N32, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N33, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.175, -DSTST, .075, N41, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.175, -DSTST, .075, N42, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.175, -DSTST, .075, N43, ANGL, 4)
    GO TO 550
540 CALL SYMBAL (.55, -DSTST, .075, N11, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.55, -DSTST, .075, N12, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.55, -DSTST, .075, N13, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.425, -DSTST, .075, N21, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N22, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.425, -DSTST, .075, N23, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.3, -DSTST, .075, N31, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N32, ANGL, 4)
    DSTST=DSTST+.267
    CALL SYMBAL (.3, -DSTST, .075, N33, ANGL, 4)
    DSTST=DSTST-.534
    CALL SYMBAL (.175, -DSTST, .075, N41, ANGL, 4)

```

```

DSTST=DSTST+.267
CALL SYMBAL(.175,-DSTST,.075,N42,ANGL,4)
DSTST=DSTST+.267
CALL SYMBAL(.175,-DSTST,.075,N43,ANGL,4)
DSTST=DSTST-.534
CALL SYMBAL(.05,-DSTST,.075,N51,ANGL,4)
DSTST=DSTST+.267
CALL SYMBAL(.05,-DSTST,.075,N52,ANGL,4)
DSTST=DSTST+.267
CALL SYMBAL(.05,-DSTST,.075,N53,ANGL,4)
550 CALL PLOTCC(0.,-DSTSP,3)
CALL PLOTCC (.75,-DSTSP,2)
IF(IMODE-1000) 800,30,30
800 N=N+1
M=0
IF(IMODE-100) 762,801,802
762 MI=MI+1
IF(MI-2) 763,764,765
763 IKEY=5250
HRST=HRSP
SPMIN=STMIN+15.
GO TO 190
764 IKEY=1160
HRSP=HRST+1.
STMIN=SPMIN
SPMIN=STMIN+15.
GO TO 190
765 IF(MI-4) 766,767,768
766 IKEY=1230
STMIN=SPMIN
HRST=HRSP
SPMIN=STMIN+30.
GO TO 190
767 IKEY=5230
HRST=HRSP
STMIN=SPMIN
SPMIN=STMIN+30.
GO TO 190
768 IF(MI-6) 769,770,30
769 IKEY=5210
HRST=HRSP
STMIN=SPMIN
SPMIN=STMIN+30.
GO TO 190
770 IF(IMODE-99) 771,772,772
771 IKEY=1170
STMIN=SPMIN
SPMIN=STMIN+480.
GO TO 190
772 IKEY=1180
STMIN=SPMIN
SPMIN=STMIN+480.
GO TO 190
801 IKEY=5100
MI=MI+1
HRST=HRSP
SPMIN=STMIN+30.
IF(MI-2) 190,30,30
802 IF(IMODE-200) 702,803,804
702 MI=MI+1
IF(MI-2) 703,704,30
763 IKEY=5100

```

```

      SPMIN=STMIN+30.
      HRST=HRSP
      GO TO 190
704  IKEY=1230
      STMIN=SPMIN+30.
      HRSP=HRST
      GO TO 190
803  MI=MI+1
      IF(MI-2) 805,806,30
805  IKEY=1100
      STMIN=SPMIN
      HRSP=HRST+1.
      GO TO 190
806  IKEY=5100
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
804  IF(IMODE-250) 853,852,852
853  MI=MI+1
      IF(MI-2) 854,855,856
854  IKEY=1100
      HRSP=HRST+1.
      STMIN=SPMIN
      GO TO 190
855  IKEY=5100
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
856  IF(MI-4) 857,858,30
857  IKEY=5230
      STMIN=SPMIN
      SPMIN=STMIN+30.
      GO TO 190
858  IKEY=1230
      STMIN=SPMIN
      SPMIN=STMIN+45.
      GO TO 190
852  IF(IMODE-300) 706,807,808
706  MI=MI+1
      IF(MI-2) 707,708,30
707  IKEY=1100
      HRSP=HRST+1.
      STMIN=SPMIN
      GO TO 190
708  IKEY=5100
      HRST=HRSP
      SPMIN=STMIN+15.
      GO TO 190
807  MI=MI+1
      IF(MI-2) 809,810,30
809  IKEY=5100
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
810  IKEY=1140
      STMIN=SPMIN
      SPMIN=STMIN+30.
      GO TO 190
808  IF(IMODE-400) 811,811,812
811  MI=MI+1
      IF(MI-2) 813,814,815
813  IKEY=1150

```

```

      HRST=HRSP
      HRSP=HRST+1.
      GO TO 190
814  IKEY=1160
      HRST=HRSP
      HRSP=HRST+1.
      GO TO 190
815  IF (MI-4) 816, 817, 818
816  IKEY=1140
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
817  IKEY=5100
      STMIN=SPMIN
      SPMIN=STMIN+15.
      GO TO 190
818  IF (MI-6) 819, 30, 30
819  IF (IMODE-350) 820, 820, 821
820  IKEY=1170
      HRSP=HRST+8.
      STMIN=SPMIN
      GO TO 190
821  IKEY=1180
      HRSP=HRST+8.
      STMIN=SPMIN
      GO TO 190
812  IF (IMODE-500) 824, 824, 1824
1825  MI=MI+1
      IF (MI-2) 1825, 30, 30
1826  IKEY=5110
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
1824  IF (IMODE-600) 1825, 1825, 1827
1827  IF (IMODE-700) 861, 1828, 741
861  MI=MI+1
      IF (MI-2) 862, 30, 30
862  IKEY=1144
      STMIN=SPMIN
      SPMIN=STMIN+15.
      GO TO 190
1828  MI=MI+1
      IF (MI-2) 1830, 30, 30
1830  IKEY=5100
      HRST=HRSP
      STMIN=SPMIN
      SPMIN=STMIN+15.
      GO TO 190
741  IF (IMODE-800) 864, 743, 742
864  MI=MI+1
      IF (MI-2) 865, 866, 867
865  IKEY=1145
      STMIN=SPMIN
      SPMIN=STMIN+30.
      GO TO 190
866  IKEY=1100
      HRSP=HRST+1.
      STMIN=SPMIN
      GO TO 190
867  IF (MI-4) 868, 869, 30
868  IKEY=1144
      HRST=HRSP

```



```

      SPMIN=STMIN+15.
      GO TO 190
869 IF(IMODE-750)30,870,870
870 IKEY=5100
      STMIN=SPMIN
      SPMIN=STMIN+10.
      GO TO 190
743 MI=MI+1
      IF(MI-2)1832,1833,30
1832 IKEY=5100
      STMIN=SPMIN
      SPMIN=STMIN+15.
      GO TO 190
1833 IKEY=1170
      STMIN=SPMIN
      HRSP=HRST+8.
      GO TO 190
742 IF(IMODE-900)714,714,1829
714 MI=MI+1
      IF(MI-2)715,716,717
715 IKEY=5250
      HRST=HRSP
      SPMIN=STMIN+15.
      GO TO 190
716 IKEY=1160
      HRSP=HRST+2.
      STMIN=SPMIN
      SPMIN=STMIN+15.
      GO TO 190
717 IF(MI-4)718,719,30
718 IKEY=5210
      HRST=HRSP
      STMIN=SPMIN
      SPMIN=STMIN+30.
      GO TO 190
719 IF(IMODE-850)744,744,745
744 IKEY=1170
      HRSP=HRST+8.0
      STMIN=SPMIN
      GO TO 190
745 IKEY=1180
      HRSP=HRST+8.
      STMIN=SPMIN
      GO TO 190
1829 IF(IMODE-1000)747,30,30
747 MI=MI+1
      IF(MI-2)748,749,750
748 IKEY=5250
      HRST=HRSP
      SPMIN=STMIN+15.
      GO TO 190
749 IKEY=1160
      STMIN=SPMIN
      HRSP=HRST+1.
      GO TO 190
750 IF(MI-4)751,752,753
751 IKEY=1230
      HRST=HRSP
      SPMIN=STMIN+30.
      GO TO 190
752 IKEY=5221
      STMIN=SPMIN

```

```

      SPMIN=STMIN*15.
      GO TO 190
753 IF (MI-6) 754, 755, 756
754 IKEY=5230
      STMIN=SPMIN
      SPMIN=STMIN*30.
      GO TO 190
755 IKEY=5210
      STMIN=SPMIN
      SPMIN=STMIN*30.
      GO TO 190
756 IF (MI-8) 757, 30, 30
757 IF (IMODE-950) 758, 758, 759
758 IKEY=1170
      STMIN=SPMIN
      HRSP=HRST+8.
      GO TO 190
759 IKEY=1180
      STMIN=SPMIN
      HRSP=HRST+8.
      GO TO 190
824 MI=MI+1
      IF (MI-2) 825, 826, 827
825 IKEY=1230
      HRST=HRSP+2.
      HRSP=HRST
      SPMIN=STMIN*30.
      GO TO 190
826 IKEY=5100
      STMIN=SPMIN
      SPMIN=STMIN*15.
      GO TO 190
827 IF (MI-4) 828, 30, 30
828 IF (IMODE-450) 829, 829, 830
829 IKEY=1170
      HRSP=HRST+8.
      STMIN=SPMIN
      GO TO 190
830 IKEY=1180
      HRSP=HRST+8.
      STMIN=SPMIN
      GO TO 190
580 K=K+1
      ICHK=K-2*(K/2)
      IF (ICLK-1) 590, 600, 590
590 CALL PLOTCC (0., -14.25, -3)
      ICM=3
      N=0
      GO TO 30
600 CALL PLOTCC (10.5, 14.25, -3)
      ICM=3
      N=0
      GO TO 30
900 CALL PLOTCC (12., 29.3, 999)
      REWIND IT20
      CALL EXIT
      END

```